Global Optimization of Integrated Objectives using Gaussian Processes

Perry Groot, Adriana Birlutiu, Tom Heskes

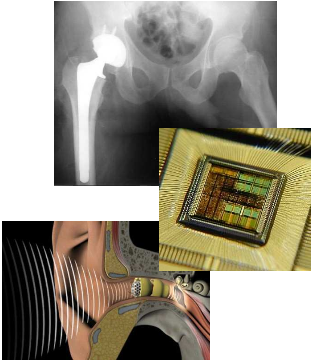
Radboud University Nijmegen
{perry,adriana,tomh}@cs.ru.nl

Machine Learning Group Colloquium 2009

Introduction

Problem: Improve off-line quality control methods (e.g., Taguchi methods) for improving a product or process to noise factors.

- Hip prosthesis [Chang, 1999]
- VLSI design [Bernardo, 1992]
- Hearing aids [Groot, ????]



Introduction

Perry Groot, Adriana Birlutiu, Tom Heskes

The HearClip project is concerned with the optimization of hearing aid parameters to optimize subject satisfaction.

A hearing aid can be interpreted as a function that has two types of input variables: $\boldsymbol{x} = (\boldsymbol{x}_c, \boldsymbol{x}_e)$ where $\boldsymbol{x}_c$ is a set of ‘manufacturing’ or control variables and $\boldsymbol{x}_e$ (e.g., amount of noise reduction) is a set of ‘environmental’ or noise variables (e.g., sounds).

The goal is find the best control variables with respect to the distribution of environmental variables.

Problem Setting

Assume the following problem setting

$$f(\mathbf{x}_c, \mathbf{x}_e) \to \mathbb{R}$$

The optimization problem can then be formulated as

$$\begin{aligned} \mathbf{x}_c^* &= \text{argmax}_{\mathbf{x}_c} E[f(\mathbf{x}_c, \mathbf{X}_e)] \\ &= \underbrace{\text{argmax}_{\mathbf{x}_c} \int_{\mathbf{x}_e} f(\mathbf{x}_c, \mathbf{x}_e) p(\mathbf{x}_e) \, \mathrm{d}\mathbf{x}_e} \end{aligned}$$

- how to represent functions?
- how to do integration?
- how to do optimization?

Gaussian Processes

A Gaussian process (GP) is collection of random variables with the property that the joint distribution of any finite subset is a Gaussian.

A GP specifies a probability distribution over functions $f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$ and is fully specified by its mean function $m(\boldsymbol{x})$ and covariance (or kernel) function $k(\boldsymbol{x}, \boldsymbol{x}')$.

Typically $m(\boldsymbol{x}) = \mathbf{0}$, which gives

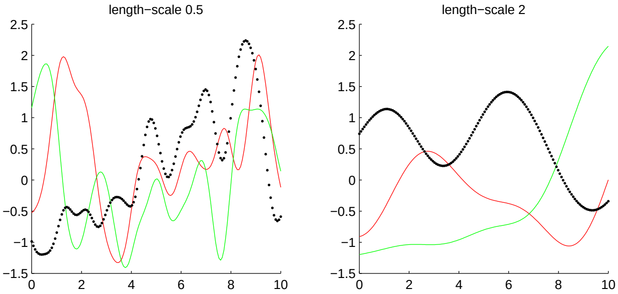
$$\{f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_l)\} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K}) \text{ with } K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Gaussian Processes - Covariance function

Squared exponential (or Gaussian) covariance function:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2\ell^2}\sum_{n=1}^{N}(x_n - x'_n)^2\right)$$

where ℓ is a length-scale parameter denoting how quickly the functions are to vary.



Gaussian Processes - Posterior process

A priori, given data $\mathcal{D} = \{X, Y\}$ with $Y = f(X)$ and test points X_* we have

$$\begin{bmatrix} f(X) \\ f(X_*) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X_*) \\ k(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

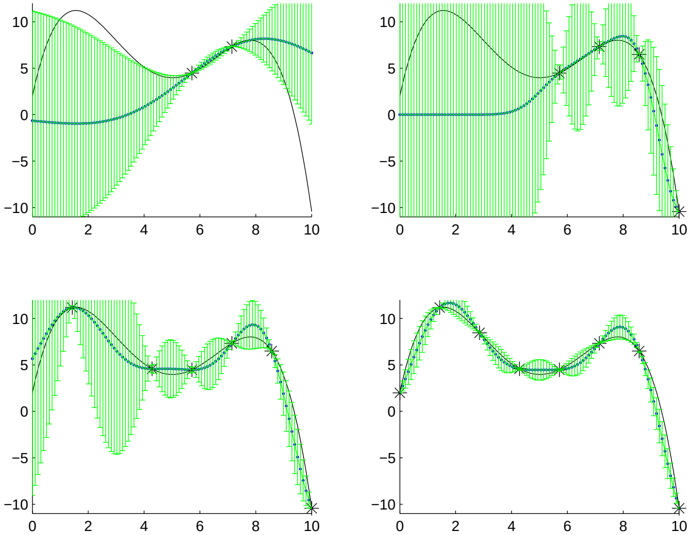
and after conditioning

$$f(X_*)|X_*, X, Y \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

with

$$\begin{aligned} \boldsymbol{\mu} &= K(X_*,X)K(X,X)^{-1}Y \\ \boldsymbol{\Sigma} &= K(X_*,X_*) - K(X_*,X)\underbrace{K(X,X)^{-1}}_{\mathcal{O}(n^3)}K(X,X_*) \end{aligned}$$

Gaussian Processes - 1D demo



Problem Setting (repeated for convenience)

Assume the following problem setting

$$f(\boldsymbol{x}_c, \boldsymbol{x}_e) \rightarrow \mathbb{R}$$

The optimization problem can then be formulated as

$$\begin{aligned} \boldsymbol{x}_c^* &= \operatorname{argmax}_{\boldsymbol{x}_c} E[f(\boldsymbol{x}_c, \boldsymbol{X}_e)] \\ &= \underbrace{\operatorname{argmax}_{\boldsymbol{x}_c} \int_{\boldsymbol{x}_e} f(\boldsymbol{x}_c, \boldsymbol{x}_e) p(\boldsymbol{x}_e) \, \mathrm{d}\boldsymbol{x}_e} \end{aligned}$$

- how to represent functions?
- how to do integration?
- how to do optimization?

Bayesian Monte Carlo

Consider the following integral

$$F = \int_{\boldsymbol{x}} f(\boldsymbol{x}) p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$$

where p is a known distribution over the inputs X.

For example,

- f could be a computer simulation needing several hours of computation to evaluate f in a single point
- $p(\boldsymbol{x})$ is the posterior distribution and $f(\boldsymbol{x})$ predictions made by the model with parameters $\boldsymbol{x}$,
- $p(\boldsymbol{x})$ is the parameter prior and $f(\boldsymbol{x}) = p(y|\boldsymbol{x})$ the likelihood (i.e., integral computes the evidence)

Bayesian Monte Carlo

Monte Carlo makes the approximation

$$F \simeq \frac{1}{T} \sum_{t=1}^{T} f(\boldsymbol{x}^{(t)})$$

with $\boldsymbol{x}^{(t)}$ random draws from $p(\boldsymbol{x})$. Disadvantages are (c.f. [O'Hagan, 1987] 'Monte Carlo is Fundamentally Unsound'):

- MC is a frequentist approach
- MC can use an irrelevant *importance sampling distribution* $q(\boldsymbol{x})$ when sampling is hard from $p(\boldsymbol{x})$
- MC ignores the values $\boldsymbol{x}^{(t)}$

Bayesian Monte Carlo

We can think of F as being random as we are uncertain about $f(\boldsymbol{x})$ because we cannot afford to compute $f(\boldsymbol{x})$ at every location.

The integral is then a Bayesian inference problem:

- put a prior on f,
- for observations, evaluate f in a number of points
- combine the prior and observations into a posterior distribution over f (which implies a distribution over F)

Bayesian Monte Carlo

When the prior f and posterior $f|\mathcal{D}$ are GPs, the distribution of F is Gaussian, $F \sim \mathcal{N}(\overline{F}, \mathrm{cov}(F))$, and is fully characterized by its mean and variance

$$\begin{aligned} \overline{F} &= \int_{\boldsymbol{x}} \overline{f}_{\mathcal{D}}(\boldsymbol{x}) p(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \\ \mathrm{cov}(F) &= \int_{\boldsymbol{x}} \int_{\boldsymbol{x}'} \mathrm{cov}(f_{\mathcal{D}}(\boldsymbol{x}), f_{\mathcal{D}}(\boldsymbol{x}')) p(\boldsymbol{x}) p(\boldsymbol{x}') \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{x}' \end{aligned} \tag{1}$$

with $\overline{f}_{\mathcal{D}}$ and $\mathrm{cov}(f_{\mathcal{D}}(\boldsymbol{x}), f_{\mathcal{D}}(\boldsymbol{x}'))$ the posterior mean and posterior variance, respectively.

Bayesian Monte Carlo - Special case

Sometimes the problem can be reduced to products of one dimensional integrals and/or some analytic expression, e.g.,

$$\begin{aligned} p(\boldsymbol{x}) &\sim \mathcal{N}(\boldsymbol{b}, \boldsymbol{B}) \\ k(\boldsymbol{x}, \boldsymbol{x}') &= w_0 \exp\left(-\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{x}')^T \boldsymbol{A}^{-1}(\boldsymbol{x}-\boldsymbol{x}')\right) \end{aligned}$$

with $\boldsymbol{A} = \mathrm{diag}(w_1^2, \ldots, w_N^2)$. Then

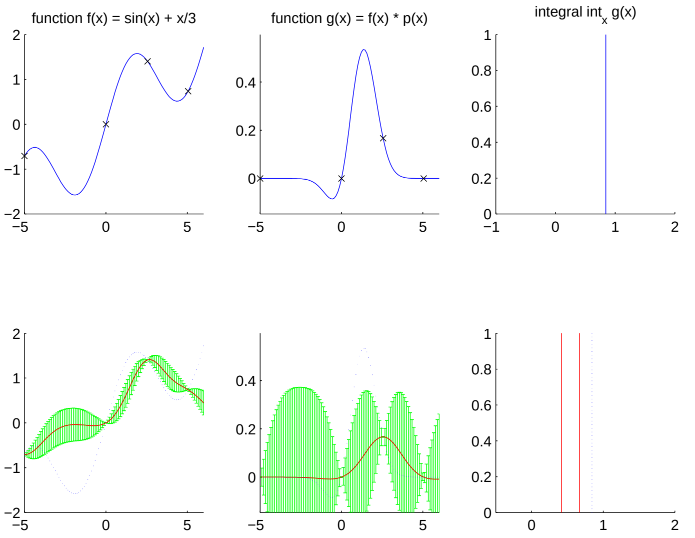
$$\overline{F} = \boldsymbol{z}\boldsymbol{K}^{-1}Y, \qquad \mathrm{cov}(F) = k_c - \boldsymbol{z}\boldsymbol{K}^{-1}\boldsymbol{z}^T$$

with

$$k_c = w_0 |2\boldsymbol{A}^{-1}\boldsymbol{B} + \boldsymbol{I}|^{-1/2}$$

$$z_l = w_0 |\boldsymbol{A}^{-1}\boldsymbol{B} + \boldsymbol{I}|^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{x}_l - \boldsymbol{b})^T(\boldsymbol{A}+\boldsymbol{B})^{-1}(\boldsymbol{x}_l - \boldsymbol{b})\right)$$

Bayesian Monte Carlo - 1D demo



Bayesian Monte Carlo - Summarizing

- Monte Carlo is a frequentist approach and technically unsound.
- Determining the integral is an inference problem when the underlying function is unknown.
- When the underlying is modeled with a Gaussian process, the integral is a Gaussian random variable.
- Some cases lead to analytic expressions.

Problem Setting (repeated for convenience)

Assume the following problem setting

$$f(\mathbf{x}_c, \mathbf{x}_e) \to \mathbb{R}$$

The optimization problem can then be formulated as

$$\begin{aligned} \mathbf{x}_c^* &= \operatorname{argmax}_{\mathbf{x}_c} E[f(\mathbf{x}_c, \mathbf{X}_e)] \\ &= \underbrace{\operatorname{argmax}_{\mathbf{x}_c} \int_{\mathbf{x}_e} f(\mathbf{x}_c, \mathbf{x}_e) p(\mathbf{x}_e) \, \mathrm{d}\mathbf{x}_e} \end{aligned}$$

- how to represent functions?
- how to do integration?
- how to do optimization?

Function Optimization

Let $f_{\max} = \max\{f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)\}$ be the best value so far. The improvement at a new point $y = f(\boldsymbol{x})$ is defined as

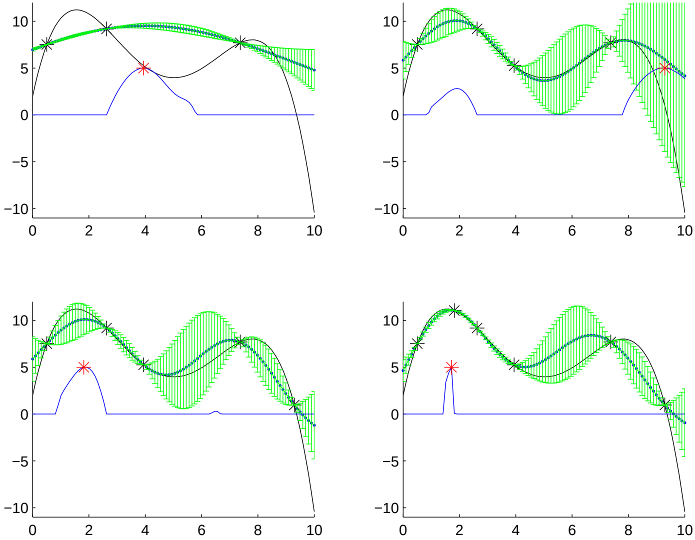
$$I(\boldsymbol{x}) = max\{0, f(\boldsymbol{x}) - f_{\max}\}$$

Using the GP prediction $y = f(\boldsymbol{x}) \sim \mathcal{N}(m, s^2)$ we obtain the Expected Improvement (EI):

$$E(I) = \begin{cases} (m - f_{\max})(1 - \Phi(d)) + s\phi(d) & s > 0 \\ 0 & s = 0 \end{cases}$$

with $d = (f_{\max} - m)/s$ and where $\Phi()$ and $\phi()$ denote the cdf and pdf of the standard normal distribution.

EI - 1D example



Generalized Expected Improvement

Advantages and disadvantages EI:

- EI allows for exploration and exploitation
- Expected Improvement can be sampled fast
- Expected Improvement often converges to a local optimum

Generalized Improvement (g positive integer):

$$I^g(\boldsymbol{x}) = \begin{cases} (f(\boldsymbol{x}) - f_{\max})^g & f(\boldsymbol{x}) > f_{\max} \\ 0 & \text{otherwise} \end{cases}$$

Larger g results in more exploration.

Applying the Theory

- Use Bayesian Monte Carlo to integrate out the environment variables
- Extend the generalized expected improvement critirion

Bayesian Monte Carlo - Integrating out $\boldsymbol{x}_e$

By integrating out only $\boldsymbol{x}_e$, the following function

$$F(\boldsymbol{x}_c) = \int_{\boldsymbol{x}_e} f(\boldsymbol{x}_c, \boldsymbol{x}_e | \mathcal{D}) p(\boldsymbol{x}_e) \, \mathrm{d}\boldsymbol{x}_e$$

becomes a Gaussian process $\mathcal{GP}(\overline{F}, \mathrm{cov}(F))$ with

$$\overline{F}(\boldsymbol{x}_c) = \boldsymbol{z}(\boldsymbol{x}_c)\boldsymbol{K}^{-1} Y$$

$$\mathrm{cov}(F(\boldsymbol{x}_c), F(\boldsymbol{x}'_c)) = k_c(\boldsymbol{x}_c, \boldsymbol{x}'_c) - \boldsymbol{z}(\boldsymbol{x}_c)\boldsymbol{K}^{-1}\boldsymbol{z}(\boldsymbol{x}'_c)^T$$

where

$$\boldsymbol{z}(\boldsymbol{x}_c)_l = w_0^{-1} k_1(\boldsymbol{x}_c, \boldsymbol{x}_{cl})\boldsymbol{z}_l$$

$$k_c(\boldsymbol{x}_c, \boldsymbol{x}'_c) = w_0^{-1} k_1(\boldsymbol{x}_c, \boldsymbol{x}'_c) k_c$$

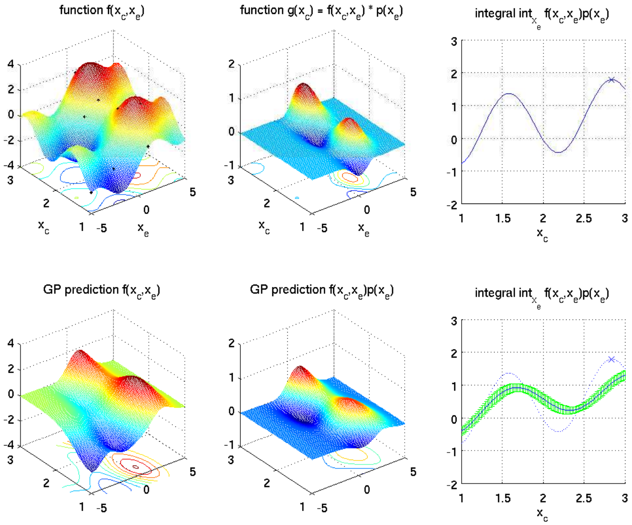
because $k((\boldsymbol{x}_c, \boldsymbol{x}_e), (\boldsymbol{x}'_c, \boldsymbol{x}'_c)) = w_0^{-1} k_1(\boldsymbol{x}_c, \boldsymbol{x}'_c) k_2(\boldsymbol{x}_e, \boldsymbol{x}'_e)$

Extending Generalized Expected Improvement

The Generalized Expected Improvement criterion is **not** directly applicable:

- The quantity $f_{\max} = \max\{F(\boldsymbol{x}_1), \ldots, F(\boldsymbol{x}_n)\}$ is no longer known – it is now a random variable.
- After applying BMC we obtain a function over the control variables $\boldsymbol{x}_c$. Expected Improvement selects a $\boldsymbol{x}_c^*$ to evaluate next, but we can only evaluate $f(\boldsymbol{x}_c^*, \boldsymbol{x}_e^*)$.

2D Demo


function f(x_c,x_e)
function g(x_c) = f(x_c,x_e) * p(x_e)
integral int_{x_e} f(x_c,x_e)p(x_e)
GP prediction f(x_c,x_e)
GP prediction f(x_c,x_e)p(x_e)
integral int_{x_e} f(x_c,x_e)p(x_e)
x_c
x_e

Experiments

Artificial data:

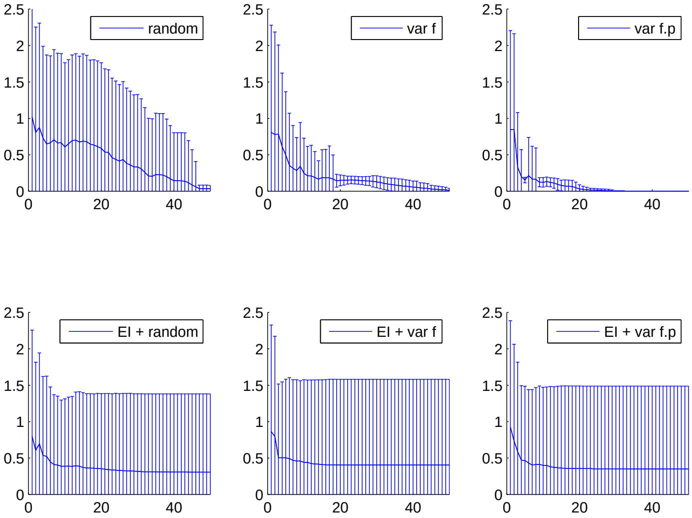
$$f(\boldsymbol{x}_c, \boldsymbol{x}_e) = \sin(\boldsymbol{x}_e) + \frac{1}{3}\boldsymbol{x}_e + \sin(5\boldsymbol{x}_c) + \frac{1}{3}\boldsymbol{x}_e - 1$$

where $-5 \leq \boldsymbol{x}_e \leq 5$, $1 \leq \boldsymbol{x}_c \leq 3$, and $p(\boldsymbol{x}_e) = \mathcal{N}(\boldsymbol{x}_e; 1, 1)$.

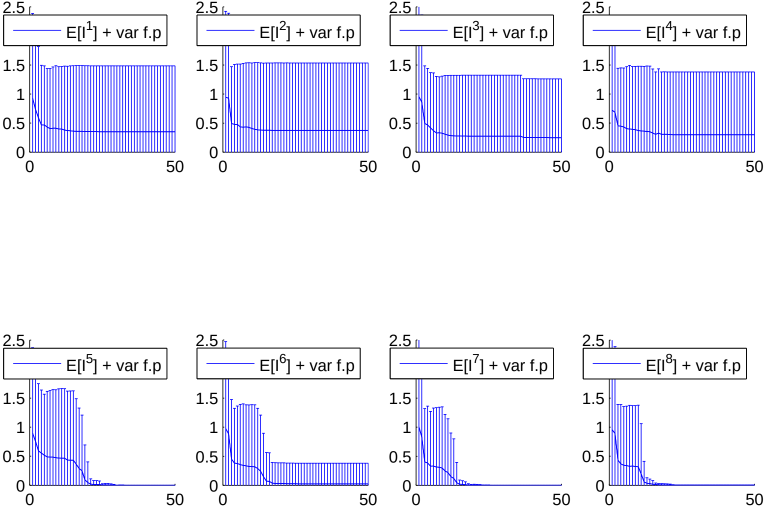
Optimization criteria:

- random selection
- maximum variance of $f(\boldsymbol{x}_c, \boldsymbol{x}_e)$.
- maximum variance of $f(\boldsymbol{x}_c, \boldsymbol{x}_e) \cdot p(\boldsymbol{x}_e)$.
- (generalized) expected improvement on $\boldsymbol{x}_c$, combined with the criteria above on $\boldsymbol{x}_e$.

Results


random
var f
var f.p
EI + random
EI + var f
EI + var f.p
2.5
2
1.5
1
0.5
0
0
20
40

Results



2.5
E[I^1] + var f.p
1.5
1
0.5
0
0
50
E[I^2] + var f.p
E[I^3] + var f.p
E[I^4] + var f.p
E[I^5] + var f.p
E[I^6] + var f.p
E[I^7] + var f.p
E[I^8] + var f.p

Future Work - Problems and Extensions

- Hyperparameter learning
- Higher dimensional problems
- Pairwise comparisons

Case Study

Suppose we want to introduce a new cake mix into the consumer market which is robust to an inaccurate setting of oven temperature and baking time.

3 Control variables: The amount of flour (F), the amount of sugar (S), and the amount of egg powder (E).

2 Noise variables: Oven temparature (T) and baking time (t).

References

[1] C.E. Rasmussen and C.K.I. Williams, Gaussian Processes for Machine Learning, 2006.

[2] C.E. Rasmussen and Z. Ghahramani, Baysian Monte Carlo, 2003.