Efficient Global Optimization of Expensive Black-Box Functions

Yifan Zhu, Gang Han, Lijin Zhang, Lingnan Yuan

Department of Statistics, Iowa State University

Table of contents

Introduction

Expensive Black-Box Function

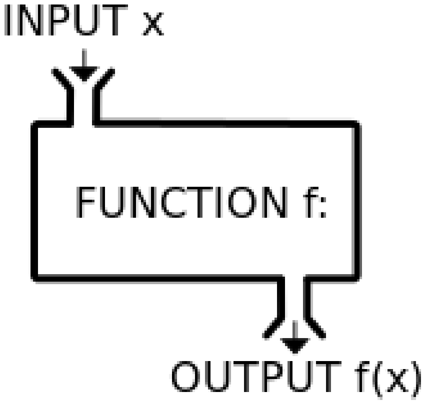

Figure 1: Black-Box Function

Main idea

- Stochastic model
- Global optimization over the response surface
- Global search using expected improvement (EI)
- Globally optimizing EI

One global optimization converted into a series of global optimizations

Stochastic Model

Model setup

$$y(x^{(i)}) = \sum_h \beta_h f_h(x^{(i)}) + \epsilon^{(i)} \qquad (i = 1, \ldots, n)$$

Problem:

- functional form
- independence

Model setup (cont.)

$$d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \sum_{h=1}^{k} \theta_i |x_h^{(i)} - x_h^{(j)}|^{p_h} \qquad (\theta_i \geq 0, p_h \in [1, 2])$$

$$Corr[\epsilon(\mathbf{x}^{(i)}), \epsilon(\mathbf{x}^{(j)})] = exp[-d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})]$$

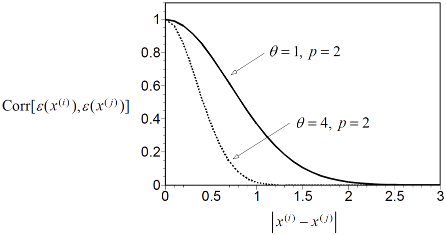

Figure 2: Example correlation function [1]

Model fitting

Design and Analysis of Computer Experiments (DACE)

$$y^{(i)} = \mu + \epsilon(\mathbf{x}^{(i)}) \qquad (i = 1, \dots, n)$$
$$\epsilon(\mathbf{x}^{(i)}) \sim N(0, \sigma^2)$$

$$L(\cdot) = \frac{1}{(2\pi)^{n/2}(\sigma^2)^{n/2}|\mathbf{R}|^{\frac{1}{2}}} exp\left[-\frac{(\mathbf{y} - \mathbf{1}\mu)'\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2}\right]$$

$$\hat{\mu} = \frac{\mathbf{1}'\mathbf{R}^{-1}\mathbf{y}}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}}, \quad \hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})'\mathbf{R}^{-1}(\mathbf{y} - \widehat{\mathbf{1}\mu)}}{n}$$

$$BLUP: \quad \hat{y}(\mathbf{x}^*) = \hat{\mu} + \mathbf{r}'\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})$$

$$MSE: \quad s^2(\mathbf{x}^*) = \hat{\sigma}^2 \left[1 - \mathbf{r}'\mathbf{R}^{-1}\mathbf{r} + \frac{(1 - \mathbf{1}'\mathbf{R}^{-}1\mathbf{r})^2}{\mathbf{1}'\mathbf{R}^{-1}\mathbf{1}}\right]$$

Example: BLUP and MSE at evaluated points

$$\hat{y}(\mathbf{x}^{(i)}) = \hat{\mu} + \mathbf{e}_i'(\mathbf{y} - \mathbf{1}\hat{\mu}) = y^{(i)}$$

$$s^2(\mathbf{x}^{(i)}) = 0$$

Comparison with other models

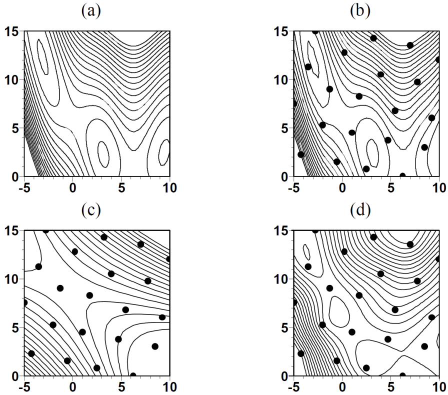

Figure 3: Illustration using Branin test function [1]

Going deeper into the model

The predictor can also be written in a form of linear combination of functions:

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \mathbf{c}'\mathbf{r} = \hat{\mu} + \sum_{i=1}^{n} c_i r_i(\mathbf{x})$$

where

$$\begin{aligned}
\mathbf{c} &= \mathbf{R}^{-1}(y - \mathbf{1}\hat{\mu}) \\
\mathbf{r}_i(\mathbf{x}) &= Corr[\epsilon(\mathbf{x}), \epsilon(\mathbf{x}^{(i)})] \quad i = 1, \ldots, n
\end{aligned}$$

Going deeper into the model (cont.)

Thin plate spine predictor:

$$\varphi(\| x - x^{(i)} \|) \quad i = 1, 2, \ldots, n$$

$$\varphi(t) = t^2 \log(t)$$

Going deeper into the model (cont.)

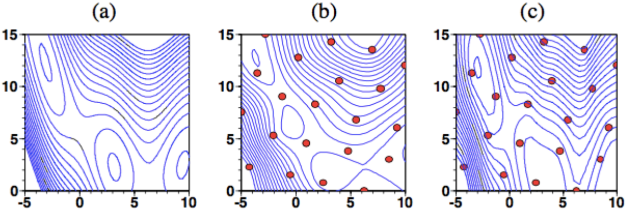

Figure 4: (a) Contours of the Branin test function; (b) a thin-plate spline fit to the 21 points; (c) a thin-plane spline fit using scaling suggested by the estimated DACE parameters.[1]

Model validation

Cross Validation

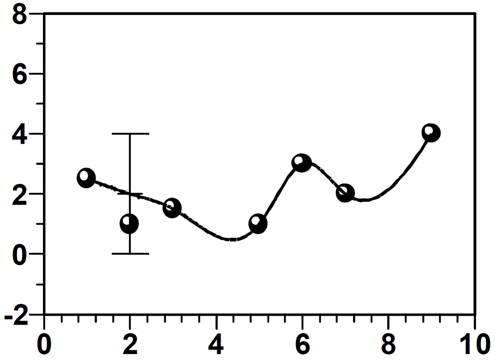

Figure 5: In ordinary cross validation, one observation (here the second) is left out and predicted back using the remaining n1 observations. The cross-validated confidence interval is the cross-validated prediction plus or minus three standard errors.[1]

Cross validation (cont.)

standardized cross-validated residual:

$$\frac{y(\mathbf{x}^i - \hat{y}_{-i}(\mathbf{x}^i))}{s_{-i}\mathbf{x}^i}$$

If the model is valid, the value should be roughly in the interval [-3,3]

Cross Validation (cont.)

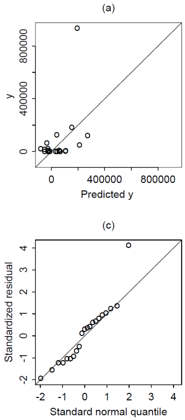

Figure 6: Diagnostic tests for the Goldstein–Price function: (a) actual function values versus cross-validated predictions; (b) standardized cross-validated residuals versus cross-validated predictions; (c) ordered standardized residuals versus standard normal quantiles.[1]

Cross validation (cont.)

Improve the fit of the model by transforming the function:

log transformation $\log(y)$, inverse transformation $-1/y$

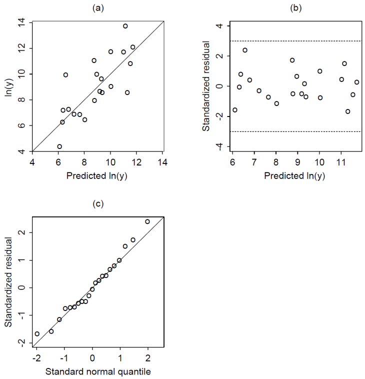

Figure 7: Diagnostic tests for the log-transformed Goldstein–Price function: (a),(b),(c) as stated in Figure 6.[1]

Global Optimization

Find the min of a fitted surface

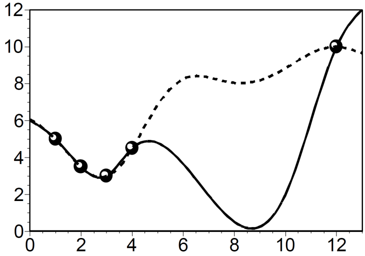

Figure 8: The solid line represents an objective function that has been sampled at 5 points shown as dots. The dotted line is a DACE predictor fit to these points.

Uncertainty about the surface

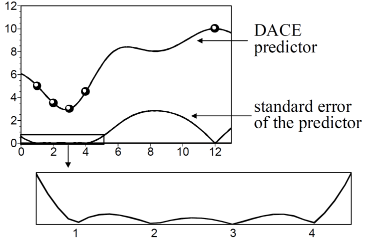

Figure 9: The DACE predictor and its standard error for a simple five-point data set.

Expected improvement

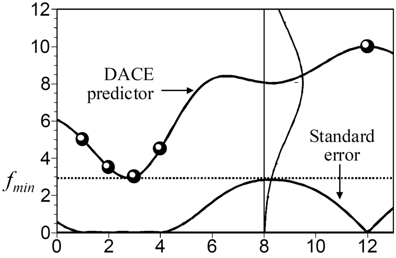

Figure 10: The uncertained value at a point can be treated as a normal random variable.

$$f_{\min} = \min(y^{(1)}, \cdots, y^{(n)}).$$

Expected improvement (cont.)

The improvement at the point x is

$$I = \max(f_{\min} - Y, 0).$$

The expected improvement is

$$E[I(x)] \equiv E[\max(f_{\min} - Y, 0)].$$

Expected improvement (cont.)

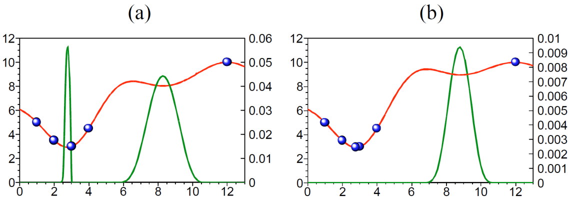

Figure 11: (a)The expected improvement function when only five points have been sampled;(b)the expected improvement after adding a point.The left scale is for the objective function and the right scale is for the expected improvement.

EGO Algorithm

1. Fit a DACE model to a set of initial points.
2. Fit the parameters of the DACE model using MLE.
3. Maximize the expected improvement.
4. If the expected improvement is less than 1 % of the best current function value, stop. Otherwise, sample the point where expected improvement is maximized, re-estimate the DACE parameters and iterate.

EGO Algorithm (cont.)

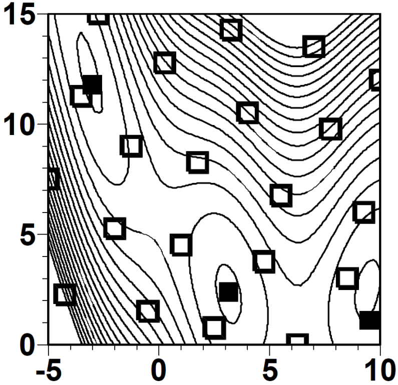

Figure 12: Example of using EGO to fit Brainin test function.

Branch-and-bound for optimal expected improvement

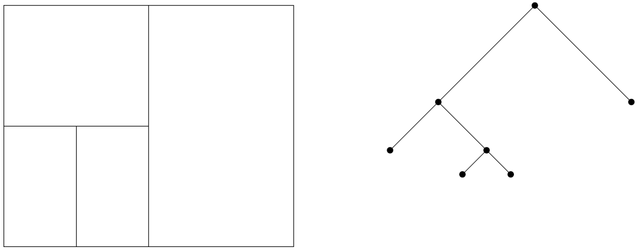

Figure 13: Branch-and Bound in $\mathbb{R}^2$

Expected improvement

$$E[I(x)] \equiv E[\max(f_{\min} - Y, 0)] \qquad \text{(conditional)}$$

Assume $Y \sim N(\hat{y}, s)$

$$E[I(x)] = (f_{\min} - \hat{y})\Phi\left(\frac{f_{\min} - \hat{y}}{s}\right) + s\phi\left(\frac{f_{\min} - \hat{y}}{s}\right)$$

Then

$$\frac{\partial E(I)}{\partial \hat{y}} = -\Phi\left(\frac{f_{\min} - \hat{y}}{s}\right) < 0$$
$$\frac{\partial E(I)}{\partial s} = \phi\left(\frac{f_{\min} - \hat{y}}{s}\right) > 0$$

Bounding $s^2(x)$ (upper bound)

$$\begin{aligned} \underset{\mathbf{r},\mathbf{x}}{\text{maximize}} \quad & \sigma^2 \left[1 - \mathbf{r}' R^{-1} \mathbf{r} + \frac{(1 - \mathbf{1}' R^{-1} \mathbf{r})^2}{\mathbf{1}' R^{-1} \mathbf{q}} \right] \\ \text{subject to} \quad & \ln(r_i) = -\sum_{h=1}^{k} \theta_h |x_h - x_h^{(i)}|^{p_h} \qquad (i = 1, \ldots, n) \\ & l_h \le x_h \le u_h \end{aligned}$$

Bounding $s^2(x)$ (upper bound) (cont.)

Equivalently,

$$
\begin{aligned}
\underset{\boldsymbol{r},\boldsymbol{x}}{\text{minimize}} \quad & -\sigma^2 \left[1 - \boldsymbol{r}' R^{-1} \boldsymbol{r} + \frac{(1 - \mathbf{1}' R^{-1} \boldsymbol{r})^2}{\mathbf{1}' R^{-1} \mathbf{1}}\right] \\
\text{subject to} \quad & \ln(r_i) + \sum_{h=1}^{k} \theta_h |x_h - x_h^{(i)}|^{p_h} \le 0 & (i = 1, \ldots, n) \\
& -\ln(r_i) + \sum_{h=1}^{k} \theta_h \left(-|x_h - x_h^{(i)}|^{p_h}\right) \le 0 & (i = 1, \ldots, n) \\
& l_h \le x_h \le u_h & (h = 1, \ldots, k) \\
& r_i^L \le r_i \le r_i^U & (i = 1, \ldots, n)
\end{aligned}
$$

Bounding $s^2(x)$ (upper bound) (cont.)

Upper left corner of Hessian:

$$2\sigma^2 \left[R^{-1} - \frac{(R^{-1}1)(R^{-1}1)'}{1'R^{-1}1} \right]$$

Need it to be non-negative definite.

Bounding $s^2(x)$ (upper bound) (cont.)

Relax the object function to make it convex:

$$-\sigma^2 \left[1 - \mathbf{r}' R^{-1} \mathbf{r} + \frac{(1 - \mathbf{1}' R^{-1} \mathbf{r})^2}{\mathbf{1}' R^{-1} \mathbf{1}}\right] + \alpha \sum_i (r_i - r_i^L)(r_i - r_i^U)$$

where

$$\alpha = \max\left\{0, -\frac{\lambda_{\min}}{2}\right\}$$

$\lambda_{\min}$ is the minimum eigenvalue the the upper left corner matrix has. In this way all eigenvalues of Hessian is non-negative, and the relaxed function is smaller than the original one.

Relax constraints by replacing $-\ln(r_i), \ln(r_i), -|x_h - x_h^{(i)}|^{p_h}$ and $|x_h - x_h^{(i)}|^{p_h}$ with linear under estimator:

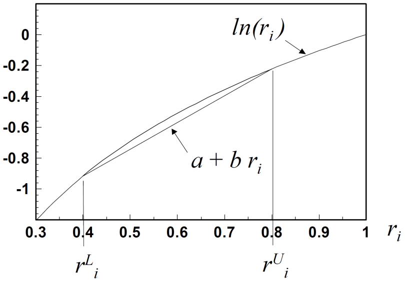

Figure 14: Linear under estimator for the nonlinear term $\ln(r_i)$. [1]

Bounding $\hat{y}(x)$ (lower bound)

Let

$$z_i(\mathbf{x}) \equiv \sum_{h=1}^{k} \theta_h \left(x_h - x_h^{(i)} \right)^2$$

Then

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \sum_{i=1}^{n} c_i r_i(\mathbf{x}) = \hat{\mu} + \sum_{i=1}^{n} c_i exp[-z_i(\mathbf{x})]$$

Bounding $\hat{y}(x)$ (lower bound) (cont.)

Find a linear under estimator for $c_i exp[-z_i]$, say $a_i + b_i z_i$, then

$$
\begin{aligned}
\hat{y}(x) &= \hat{\mu} + \sum_{i=1}^{n} c_i exp[-z_i(x)] \\
&\geq \hat{\mu} + \sum_{i=1}^{n} [a_i + b_i z_i(x)] \\
&= \hat{\mu} + \sum_{i=1}^{n} \left[a_i + b_i \sum_{h=1}^{k} \theta_h \left(x_h - x_h^{(i)} \right)^2 \right] \\
&= \hat{\mu} + \sum_{i=1}^{n} a_i + \sum_{h=1}^{k} \sum_{i=1}^{n} b_i \left(x_h - x_h^{(i)} \right)^2
\end{aligned}
$$

For each h, $\sum_{i=1}^{n} b_i \left(x_h - x_h^{(i)} \right)^2$ is a one-dimension quadratic form which can be easily minimized over $x_h \in [l_h, u_h]$.

Questions?

References

D. R. Jones, M. Schonlau, and W. J. Welch.
Efficient global optimization of expensive black-box functions.
Journal of Global optimization, 13(4):455–492, 1998.