GLOBAL OPTIMIZATION WITH NONPARAMETRIC FUNCTION FITTING

Matthias Schonlau, and William J. Welch, University of Waterloo
Matthias Schonlau, Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario N2L 3G1 Canada

Key Words: Computer Experiments, Global Optimization, Stochastic Processes

1 Introduction

Global optimization, that is the search for a global extremum, is a problem commonly encountered in practice. Sometimes it is extremely costly to evaluate a function at a design point x, as is the case for example at Boeing:

> "Designing helicopter blades to achieve low vibration is an extreme example of a problem where it is prohibitively expensive to compute responses for large numbers of design alternatives." (Siam News, Jan/Feb 1996)

In order to find the global minimum, one is then interested in minimizing the total number of function evaluations needed to do so. We propose here an algorithm aimed at minimizing the total number of function evaluations for finding the global extremum of a deterministic function. The amount of computation that it takes to decide on design points is not a concern.

With this objective in mind, we replace the unknown function by a stochastic model estimated from previous function evaluations. We then choose design points based on the model, rather than just on the last function evaluation.

The outline of this paper is as follows. In section 2 we briefly discuss a common stochastic process model used in the analysis of computer experiments. Section 3 describes the minimization algorithm that we employ. In Section 4 we illustrate the minimization algorithm by means of two simple examples.

2 A Stochastic Process Model

We will briefly review a common stochastic process model used in the analysis of computer experiments that will be needed later on.

The data from a computer experiment consist of n vectors of covariate values (or inputs) denoted by $\mathbf{x}_1, \ldots, \mathbf{x}_n$ for the k covariates $x_1, \ldots, x_k$ as specified by a particular experimental design. The corresponding response values (or outputs) are denoted $\mathbf{y} = (y_1, \ldots, y_n)^t$. Then, following the approach of, e.g., Welch et al. (1992), the response is treated as a realization of a stochastic process:

$$Y(\mathbf{x}) = \beta + Z(\mathbf{x}),$$

where $E(Z(\mathbf{x})) = 0$ and $\mathrm{Cov}(Z(\mathbf{w}), Z(\mathbf{x})) = \sigma^2 R(\mathbf{w}, \mathbf{x})$ for two inputs $\mathbf{w}$ and $\mathbf{x}$. The correlation function $R(\cdot, \cdot)$ can be tuned to the data, which for this paper is assumed to have the form:

$$R(\mathbf{w}, \mathbf{x}) = \prod_{j=1}^{k} \exp(-\theta_j |w_j - x_j|^{p_j}), \qquad (1)$$

where $\theta_j \geq 0$ and $0 < p_j \leq 2$. The p_j's can be interpreted as smoothness parameters (smoother as the p's increase) which indicate the smoothness of the response surface and the θ's indicate how local the predictor is (more local as the θ's increase).

The best linear unbiased predictor of y at an untried $\mathbf{x}$ can be shown to be:

$$\hat{y}(\mathbf{x}) = \hat{\beta} + \mathbf{r}^t(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\beta}), \qquad (2)$$

where $\mathbf{r}^t(\mathbf{x})$ is the vector of the correlations between $\mathbf{x}$ and each of the n design points, $\hat{\beta}$ is the generalized least squares estimator of β, $\mathbf{R}$ is the correlation matrix with elements defined by (1) and $\mathbf{1}$ is a vector of 1's.

The MSE of the estimate can be derived as:

$$\mathrm{MSE}[\hat{y}(\mathbf{x})] = \sigma_z^2 \left[1 - (1 \; \mathbf{r}_x^t) \begin{pmatrix} 0 & \mathbf{1}^t \\ \mathbf{1} & \mathbf{R} \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ \mathbf{r}_x \end{pmatrix} \right] \qquad (3)$$

The predictor in (2) has proven to be accurate for numerous applications, see e.g. Currin et al. (1991), Sacks et al. (1989a), Sacks et al. (1989b), Welch et al. (1992).

3 Expected Improvement Algorithm

In this section we give a heuristic algorithm for a sequential design strategy for detecting the global minimum of a deterministic function. We will call this algorithm the *expected improvement algorithm*. We assume without loss of generality that the extremum of interest is a minimum. A maximization problem can be easily turned into a minimization problem by multiplying the function by (-1).

The expected improvement algorithm proceeds in two steps:

1. First we sample n points of the true function in a space filling manner. Latin hypercube sampling schemes are particularly suitable here.

2. We then proceed sequentially sampling one point at a time. At each step we sample the point with the greatest expected improvement over the current minimal sampled function value. The derivation of the expected improvement is given below.

   After each sampling step the predictor is updated, and the expected improvement is recalculated.

After some preliminaries, we will now define the improvement I and derive the expected improvment $E(I)$.

Suppose we have fitted the BLUP in equation (2) to the data accumulated at some stage. To predict $Y(\mathbf{x})$ at an untried $\mathbf{x}$, we have $\hat{y}(\mathbf{x})$ with a mean squared error given by (3). For notational simplicity, we omit the dependence on $\mathbf{x}$, and denote $\hat{y}(\mathbf{x})$ by $\hat{\mu}$ and the root mean squared error by $\hat{\sigma}$. Next, we take the distribution of the unknown $Y = Y(\mathbf{x})$ as $N(\hat{\mu}, \hat{\sigma}^2)$.

Definition: If the function is sampled at $\mathbf{x}$ to determine $y = y(\mathbf{x})$ then the improvement I over f_{min}, the minimal sampled function value so far, is defined as

$$I = \begin{cases} f_{min} - y & y < f_{min} \\ 0 & \text{otherwise} \end{cases}$$

where f_{min} denotes the current minimal sampled function value.

We can rewrite the improvement as

$$I = \begin{cases} \hat{\sigma}(f'_{min} - z) & z < f'_{min}, \hat{\sigma} > 0 \\ 0 & otherwise \end{cases}$$

where $z = \frac{y-\hat{\mu}}{\hat{\sigma}}$ and $f'_{min} = \frac{f_{min}-\hat{\mu}}{\hat{\sigma}}$.

Using the assumed normal distribution for the unknown Y, in the case of $\hat{\sigma} > 0$ the expected improvement is

$$\begin{aligned} E(I) &= \hat{\sigma} \int_{-\infty}^{f'_{min}} (f'_{min} - z)\phi(z)dz \\ &= \hat{\sigma} \left[f'_{min}\Phi(f'_{min}) + \left[\frac{e^{-z^2/2}}{\sqrt{2\pi}} \right]_{-\infty}^{f'_{min}} \right] \\ &= \hat{\sigma}[f'_{min}\Phi(f'_{min}) + \phi(f'_{min})] \\ &= (f_{min} - \hat{\mu})\Phi(\frac{f_{min} - \hat{\mu}}{\hat{\sigma}}) + \hat{\sigma}\phi(\frac{f_{min} - \hat{\mu}}{\hat{\sigma}}). \end{aligned} \tag{4}$$

In the case of $\sigma = 0$, the improvement is zero and consequently so is the expected improvement. Hence, in summary

$$E(I) = \begin{cases} (f_{min} - \hat{\mu})\Phi(\frac{f_{min}-\hat{\mu}}{\hat{\sigma}}) + \hat{\sigma}\phi(\frac{f_{min}-\hat{\mu}}{\hat{\sigma}}) & \hat{\sigma} > 0 \\ 0 & \hat{\sigma} = 0 \end{cases} \tag{5}$$

Note that improvement is nonnegative and hence the expected improvement is nonnegative.

The expected improvement will tend to be large at a point whose predicted value is very small or where there is a lot of uncertainty associated with the current predicted $Y(\mathbf{x})$ at that point.

A practical problem, though, is finding the global maximum of the expected improvement over a continuous region. Currently, we start local searches at each of the design points.

4 Examples

In this section, we demonstrate graphically the performance of the expected improvement algorithm by means of two relatively simple examples.

The Branin function (Jones et al., 1993) is

$$f(x_1, x_2) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10. \quad (6)$$

The x ranges are $-5 \leq x_1 \leq 10$ and $0 \leq x_2 \leq 15$. This function is challenging as it has three global minima. Because the function has only two x variables it is especially suitable for visualization.

Initially, we sample the function at 21 points generated by a Maximin design within the class of Latin Hypercubes (Welch, work in progress). We then employ the expected improvement algorithm outlined in section 2. The design of the initial 21 points (denoted by a dot) and the following points resulting from the sequential optimization (denoted by their respective number) can be seen in Figure 1.

We can see that following the initial design points, sampled points cluster around the three global optima. For this example we use a stopping criterion based on the size of the expected improvement relative to the current minimum. Work on an alternate stopping criterion is in progress.

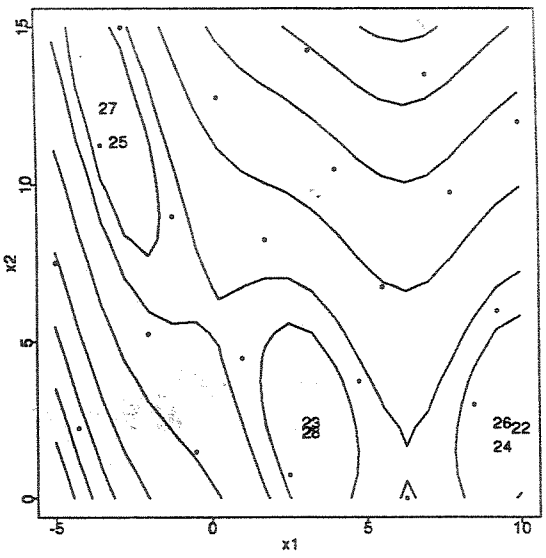

Figure 1: Design and Sequential Minimization Design for the Branin Function. Sequential Minimization Design Points are Labelled in Order of Appearance.

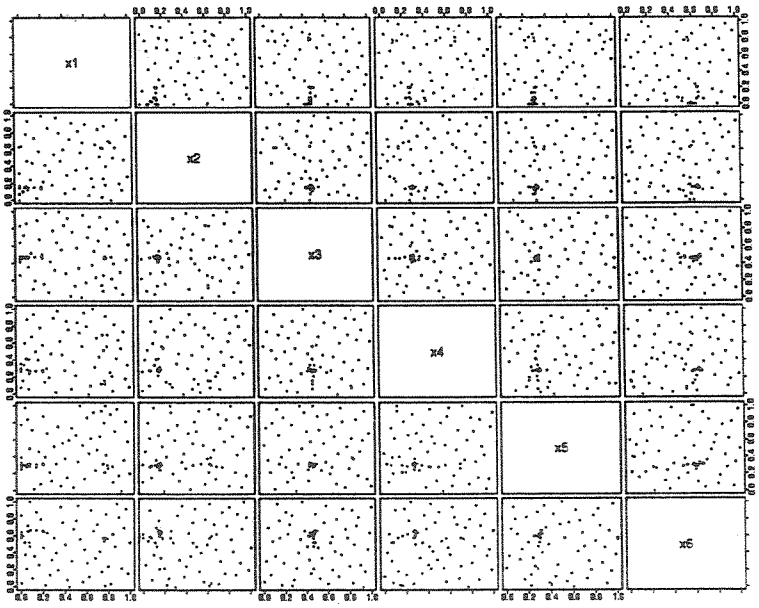

Figure 2: Two Dimensional Projections of the 51 Design Points and the 28 Sequential Minimization Design Points for the Hartman Function.

The Hartman function (Törn and Žilinskas, 1987) is given as

$$f(x_1, \ldots, x_6) = -\sum_{i=1}^{4} c_i exp\left[-\sum_{j=1}^{6} \alpha_{ij}(x_j - p_{ij})^2\right] \quad (7)$$

where c_i, p_{ij} and α_{ij} are coefficients. They will not be reproduced here due to space constraints (see Törn and Žilinskas, 1987). The x ranges are $0 \leq x_i \leq 1$ for $i = 1, \ldots, 6$. For the initial design we choose 51 points. Roughly speaking, we use about 10 points for each active variable. Numbers like 11, 21, 51 result in convenient design points, but any other number could be chosen, too. We then use the expected improvement algorithm to determine the minimum. Two dimensional projections of the design including the design points resulting from the minimization can be seen in Figure 2 . We can see that the design clusters around one single point which indeed is the minimum. During the minimization, only 28 additional points were sampled.

References

[1] Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991), "Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments," *JASA*, 86, 953-963.

[2] Jones, D.R., Perttunen, C.D., and Stuckman, B.E. (1993), "Lipschitzian Optimization Without the Lipschitz Constant", *Journal of Optimization Theory and Application*, 79, 157-181.

[3] Sacks, J., Schiller, S.B., and Welch, W.J. (1989), "Designs for Computer Experiments," *Technometrics*, 31, 41–47.

[4] Sacks, J., Welch, W.J., Mitchell, T.J., and Wynn, H.P. (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, 4, 409–435.

[5] Törn, A. and Žilinskas, A. (1987), "Global Optimization", Springer Verlag, Berlin.

[6] Welch, W.J., Buck, R.J., Sacks, J., Wynn, H.P. , Mitchell, T.J., and Morris, M.D. (1992), "Screening, Predicting, and Computer Experiments," *Technometrics*, 34, 15–25.