7

Design of Computer Experiments for Optimization, Estimation of Function Contours, and Related Objectives

Derek Bingham

Simon Fraser University, Burnaby, BC

Pritam Ranjan

Acadia University, Wolfville, NS

William J. Welch

University of British Columbia, Vancouver, BC

7.1 Introduction

A computer code or simulator is a mathematical representation of a physical system, for example a set of differential equations. Such a simulator takes a set of input values or conditions, $\mathbf{x}$, and from them produces an output value, $y(\mathbf{x})$, or several such outputs. For instance, one application we use for illustration simulates the average tidal power, y, generated as a function of a turbine location, $\mathbf{x} = (x_1, x_2)$, in the Bay of Fundy, Nova Scotia (Ranjan et al., 2011). Performing scientific or engineering experiments via such a computer code (i.e., a computer experiment) is often more time and cost effective than running a physical experiment or collecting data directly.

A computer experiment may have objectives similar to those of a physical experiment. For example, computer experiments are often used in manufacturing or process development. If y is a quality measure for a product or process, an experiment could aim to optimize y with respect to $\mathbf{x}$. Similarly, an experiment might aim to find sets or contours of $\mathbf{x}$ values that make y equal a specified target value — a type of inverse problem. Such scientific and engineering objectives are naturally and efficiently achieved via so-called data-adaptive sequential design, which we describe below. Essentially, each new run (i.e., new set of input values) is chosen based on the analysis of the data so far, to make the best expected improvement in the objective. In a computer experiment, choosing new experimental runs, restarting the exper-

iment, etc. pose only minor logistical challenges if these decisions are also computer-controlled, a distinct advantage relative to a physical experiment.

Choosing new runs sequentially for optimization, moving y to a target, etc. has been formalized using the concept of expected improvement (Jones et al., 1998). The next experimental run is made where the expected improvement in the function of interest is largest. This expectation is with respect to the predictive distribution of y from a statistical model relating y to $\mathbf{x}$. By considering a set of possible inputs $\mathbf{x}$ for the new run, we can choose that which gives the largest expectation.

We illustrate this idea with two examples in Section 7.2. Then we describe formulations of improvement functions and their expectations in Section 7.3. Expectation implies a statistical model, and in Section 7.4 we outline the use of Gaussian process models for fast emulation of computer codes. In Section 7.5 we describe some extensions to other, more complex scientific objectives.

7.2 Expected Improvement and Sequential Design: Basic Ideas

The ideas behind expected improvement and data-adaptive sequential design are illustrated via two examples. The first, a tidal-power application, shows the use of expected improvement in sequential optimization. In the second example, we use a simulator of volcanic pyroclastic flow to illustrate how to map out a contour of a function.

7.2.1 Optimization

Ranjan et al. (2011) described output from a 2D computer-model simulation of the power produced by a tidal turbine in the Minas Passage of the Bay of Fundy, Nova Scotia. In this simplified version of the problem there are just two inputs for the location of a turbine. Originally, the input space was defined by latitude-longitude coordinates for a rectangular region in the Minas Passage (see Figure 5 of Ranjan et al., 2011). The coordinates were transformed so that x_1 is in the direction of the flow and x_2 is perpendicular to the flow. Furthermore, only an interesting part of the Minas Passage was considered, with $x_1 \in [.75, .95]$ and $x_2 \in [.2, .8]$. The computational model generates an output, y, the extractable power in MW, averaged over a tidal cycle for inputs (x_1, x_2). For the simplified demonstration here, y was computed for 533 runs on a 13×41 grid of x_1 and x_2 values, which produced the contour plot of Figure 7.1(a).

We now demonstrate how the turbine location optimizing the power (i.e., $\max y(x_1, x_2)$) can be found with far fewer than 533 runs of the computer code. Such an approach would be essential for the computer experiment of

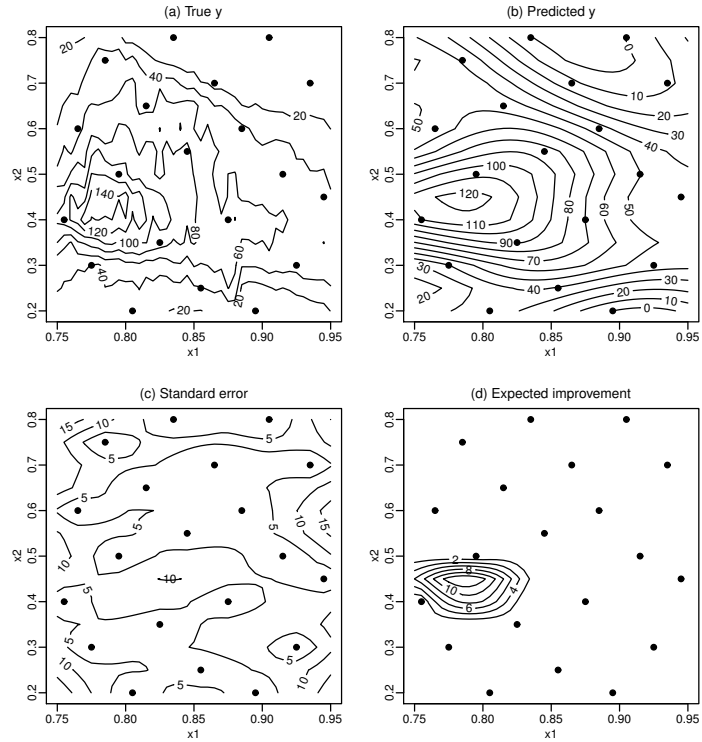

FIGURE 7.1: Initial 20-run design and analysis for the tidal-power application: (a) true power, y, in MW; (b) predicted power, $\hat{y}(\mathbf{x})$; (c) standard error of prediction, $s(\mathbf{x})$; and (d) expected improvement, $\mathrm{E}\{I(\mathbf{x})\}$. The design points from an initial 20-run maximin Latin hypercube are shown as filled circles. All plots are functions of the two input variables, x_1 and x_2, which are transformed longitude and latitude.

ultimate interest. A more realistic computer model has a grid resolution 10 times finer in each coordinate and introduces vertical layers in a 3D code. The running time would be increased by several orders of magnitude. Moreover, the final aim is to position several turbines, which would interfere with each other, and so the optimization space (or input space) is larger than two or three dimensions. Thus, the ultimate goal is to optimize a high-dimensional function with a limited number of expensive computer model runs. Inevitably, much of the input space cannot be explicitly explored, and a statistical approach to predict outcomes (extractable power) along with an uncertainty measure is required to decide where to make runs and when to stop. The expected improvement criterion addresses these two requirements.

Consider a more limited computer experiment with just 20 runs, as shown by the points in Figure 7.1. The experimental design (i.e., the locations of the 20 points) is a maximin Latin hypercube (Morris and Mitchell, 1995), a stratified scheme that is "space-filling" even in higher dimensions. The choice of 20 runs is based on the heuristic rule that an initial computer experiment has $n = 10d$ observations (Loeppky et al., 2009), where d is the input dimension; here $d = 2$. Among the 20 initial runs, the largest y observed, denoted by $y_{\max}^{(20)}$, is 109.7 MW at $(x_1, x_2) = (.755, .4)$. The expected improvement algorithm tries to improve on the best value found so far, as new runs are added.

At each iteration of the computer experiment we obtain a predictive distribution for $y(\mathbf{x})$ conditional on the runs so far. This allows prediction of the function at input vectors $\mathbf{x}$ where the code has not been run. A Gaussian process (GP) statistical model is commonly used for prediction, as outlined in Section 7.4, though this is not essential. A GP model was fit here to the data from the first 20 runs, giving the point-wise predictions, $\hat{y}(\mathbf{x})$, of $y(\mathbf{x})$ in Figure 7.1(b) along with standard errors of prediction, $s(\mathbf{x})$, in Figure 7.1(c). The standard error is a statistical measure of uncertainty concerning the closeness of the predicted value to the actual true value of $y(\mathbf{x})$. We show the predicted values and standard errors through contours in Figures 7.1(b) and 7.1(c).

Figures 7.1(b) and (c) are informative about regions in the input space that are promising versus unpromising for further runs of the code. While the $\hat{y}(\mathbf{x})$ prediction surface is nonlinear, it suggests there is a single, global optimum. Moreover, the $s(\mathbf{x})$ surface is uniformly below about 15 MW: For much of the input space, $\hat{y}(\mathbf{x})$ is so much smaller than $y_{\max}^{(20)}$ relative to $s(\mathbf{x})$ that a new run is expected to make virtually zero improvement.

The expected improvement (EI) for a candidate new run at any $\mathbf{x}$ is computed from the predictive distribution of $y(\mathbf{x})$; see Section 7.3.1 for the formal definition of EI. Figure 7.1(d) shows the EI surface based on predictive distributions from a GP that was fit to the data from the initial 20 runs of the tidal-power code. By the definition in Section 7.3.1, the improvement can never be negative (if the output from the new run does beat the current optimum, the current optimum stands). Consequently, the EI is always non-negative too. Figure 7.1(d) indicates that for most of the input space EI is near zero and a new run would be wasted, but there is a sub-region where EI is more than 12 MW. Evaluating EI over the 13×41 grid shows that the maximum EI is 13.9 MW at $\mathbf{x} = (.785, .45)$. In other words, a new code run to evaluate $y(.785, .45)$ is expected to beat $y_{\max}^{(20)} = 109.7$ MW by 13.9 MW.

Thus, run 21 of the sequential design for the computer experiment is at $\mathbf{x}^{(21)} = (.785, .45)$. The actual power obtained from the simulator is $y = 159.7$ MW, so the best y found after 21 runs is $y_{\max}^{(21)} = 159.7$ MW, and this is the value to beat at the next iteration. Note that the actual improvement in the optimum from the new run is $159.7 - 109.7 = 50.0$ MW, compared with an expectation of about 13.9 MW.

The new run raises concerns about the statistical model. Before making the new run, the predictive distribution of $y(\mathbf{x}^{(21)})$ is approximately Normal, specifically $\mathcal{N}(123.5, 5.67^2)$, an implausible distribution given the large value of the standardized residual

$$\frac{y(\mathbf{x}^{(21)}) - \hat{y}(\mathbf{x}^{(21)})}{s(\mathbf{x}^{(21)})} = \frac{159.7 - 123.5}{5.67} = 6.4.$$

One deficiency is that $s(\mathbf{x})$ may not reflect all sources of uncertainty in estimation of the parameters of the GP (see Section 7.4). A more important reason here, however, is that the new observation successfully finds a peak in the input space, a sub-region where the output function is growing rapidly and uncertainty is larger. In contrast, the first 20 runs were at locations where the function is flatter and easier to model. The GP model fit to the initial runs underestimated the uncertainty of prediction in a more difficult part of the input space.

Careful consideration of the properties of a GP model and the possible need for transformations is particularly relevant for sequential methods based on predictive distributions. Uncertainty of prediction is a key component of the EI methodology, so checking that a model has plausible standard errors of prediction is critical.

One way of improving the statistical emulator of the tidal-power code is to consider transformation of the output. This is described in the context of the volcano example of Section 7.2.2, where transformation is essential. For the tidal-power example, persisting with the original model will show that it adapts to give more plausible standard errors with a few more runs.

The GP model and predictive distributions are next updated to use the data from all 21 runs now available. Figure 7.2(a) shows the location of the new run as a "+" and the updated $\hat{y}(\mathbf{x})$. Similarly, Figure 7.2(b) gives the updated $s(\mathbf{x})$. A property of the GP fit is that $s(\mathbf{x})$ must be zero at any point $\mathbf{x}$, where $y(\mathbf{x})$ is in the dataset for the fit; see Jones et al. (1998) for a derivation of this result. Thus, $s(\mathbf{x})$ is zero at the new run, and Figure 7.2(b) shows it is less than 5 MW near the new run. Comparing with Figure 7.1(c), it is seen that $s(\mathbf{x})$ was 5 MW or more in this neighborhood for the GP fit before the new run.

On the other hand, comparison of Figures 7.1(c) and 7.2(b) shows that $s(\mathbf{x})$ has *increased* outside the neighborhood of the new run. For example, at the right edge of Figure 7.1(c), $s(\mathbf{x})$ barely reaches 15 MW, yet $s(\mathbf{x})$ often exceeds 15 MW or even 20 MW at the same locations in Figure 7.2(b). The 21-run GP fit has adapted to reflect the observed greater sensitivity of the output to x_1 and x_2. (For instance, the estimate of the GP variance parameter σ^2, defined in Section 7.4, increased.) Thus, the model has at least partially self corrected and we continue with it.

The EI contour plot in Figure 7.2(c) suggests that there is little further improvement to be had from a further run anywhere. If a run number 22 is made, however, it is not located where $\hat{y}(\mathbf{x})$ is maximized; that location co-

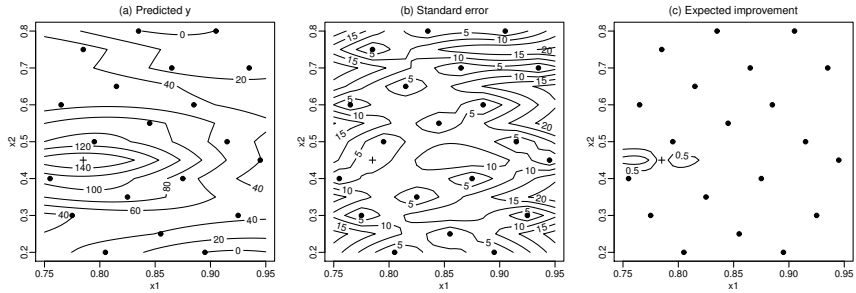

FIGURE 7.2: Analysis of the tidal-power application after 21 runs: (a) predicted power, $\hat{y}$; (b) standard error of prediction, $s(\mathbf{x})$; and (c) expected improvement, $\mathrm{E}\{I(\mathbf{x})\}$. The new design point is shown as a "+."

incides with run 21 and there would be no gain. Rather, the maximum EI of about 1.3 MW occurs at a moderate distance from the location providing maximum $\hat{y}(\mathbf{x})$. As we move away from run 21, the standard error of prediction increases from zero until it is large enough to allow a modest expected improvement. Thus, this iteration illustrates that EI trades off local search (evaluate where $\hat{y}(\mathbf{x})$ is optimized) and global search (evaluate where uncertainty concerning fitted versus actual output values, characterized by $s(\mathbf{x})$, is optimized).

With this approach, EI typically indicates smaller potential gains as the number of iterations increases. Eventually, the best EI is deemed small enough to stop. It turns out that run 21 found the global maximum for extractable power on the 13×41 grid of locations.

7.2.2 Contour Estimation

We illustrate sequential design for mapping out a specified contour of a computer-model function using TITAN2D computer model runs provided by Elaine Spiller. They relate to the Colima volcano in Mexico. Again for ease of illustration, there are two input variables: x_1 is the pyroclastic flow volume (log base 10 of m^3) of fluidized gas and rock fragments from the eruption; and x_2 is the basal friction angle in degrees, defined as the minimum slope for the volcanic material to slide. The output z is the maximum flow height (m) at a single, critical location. As is often the case, the code produces functional output, here flow heights over a 2D grid on the earth's surface, but the output for each run is reduced to a scalar quantity of interest, the height at the critical location.

Following Bayarri et al. (2009), the scientific objective is to find the values of x_1 and x_2 where $z = 1$, a contour delimiting a "catastrophic" region. Bayarri et al. (2009) used the same TITAN2D code but for a different volcano. They

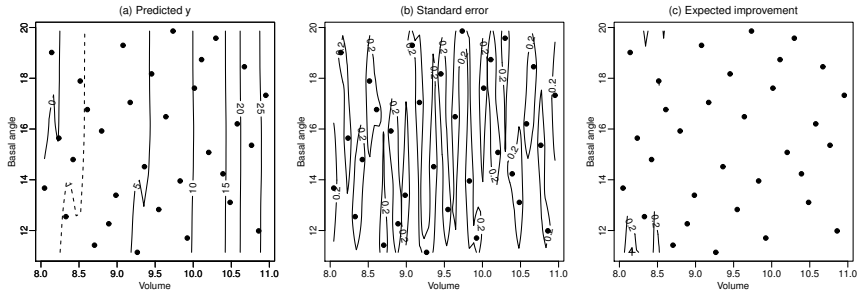

FIGURE 7.3: Analysis of the initial 32-run design for the volcano application: (a) predicted height, $\hat{y}(\mathbf{x})$, where $y = \sqrt{z}$; (b) standard error of prediction, $s(\mathbf{x})$; and (c) expected improvement, $\mathrm{E}\{I(\mathbf{x})\}$. The design points of the initial 32-run design are shown as filled circles. The new design point chosen by the EI criterion is shown as a "+" in the lower left corner of (c).

also conducted their sequential experiment in a less formal way than in our illustration of the use of EI.

There are 32 initial runs of the TITAN2D code. They are located at the points shown in Figure 7.3(a). The predicted flow height surface also shown in Figure 7.3(a) relates to a GP model fit to the transformed simulator output $y = \sqrt{z}$. This choice was made by trying GP models on three different scales: the z untransformed height; $\log(z + 1)$, as chosen by Bayarri et al. (2009); and $\sqrt{z}$. Our final choice of $y = \sqrt{z}$ results from inspection of standard cross-validation diagnostics for GP models (Jones et al., 1998).

The dashed curve in Figure 7.3(a) shows the contour where $\hat{y}(\mathbf{x}) = 1$. This maps out the contour of interest in the (x_1, x_2) input space, but it is based on predictions subject to error. The standard errors in Figure 7.3(b) are substantial, and sequential design via EI aims to improve the accuracy of the estimate of the true $y(\mathbf{x}) = 1$ contour.

The EI criterion adapted for the contouring objective is defined in Section 7.3.2. It is computed for the initial design of the volcano example in Figure 7.3(c). EI suggests improving the accuracy of the contour by taking the next run at $(x_1, x_2) = (8.2, 11.1)$. Inspection of Figures 7.3(a) and 7.3(b) show that this location is intuitively reasonable. It is in the vicinity of the predicted $\hat{y}(\mathbf{x}) = 1$ contour and has a relatively large predictive standard error. Reducing substantial uncertainty in the vicinity of the estimated contour is the dominant aspect of the EI measure of Equation (7.4).

The tidal-flow and volcano applications both have a 2D input space for ease of exposition, but the same approaches apply to higher dimensions, where choosing runs in a sequential design would be more problematic with ad hoc methods.

7.3 Expected Improvement Criteria

In this section, we briefly define the improvement and EI in general. We then review two implementations, specific to global optimization and contour estimation, respectively.

Let $I(\mathbf{x})$ be an improvement function defined for any $\mathbf{x}$ in the input space, χ. The form of the improvement depends on the scientific objective, such as improving the largest y found so far in maximization. In general, it is formulated for efficient estimation of a pre-specified computer-model output feature, $\psi(y)$. Typically, before taking another run, $I(\mathbf{x})$ is an unobserved function of $\mathbf{x}$, the unknown computer-model output $y(\mathbf{x})$, the predictive distribution of $y(\mathbf{x})$, and the best estimate so far of $\psi(y)$.

Given a definition of $I(\mathbf{x})$, as its name suggests, the corresponding EI criterion is given by the expectation of $I(\mathbf{x})$, viz.

$$\mathrm{E}\{I(\mathbf{x})\} = \int I(\mathbf{x}) f(y|\mathbf{x})\, \mathrm{d}y.$$

Here expectation is with respect to the predictive distribution of $y(\mathbf{x})$ conditional on all runs so far, $f(y|\mathbf{x})$. Assuming the sequential design scheme selects one new input point at a time, the location of the new point, $\mathbf{x}^{\mathrm{new}}$, is the global maximizer of $\mathrm{E}\{I(\mathbf{x})\}$ over $\mathbf{x} \in \chi$.

7.3.1 EI for Global Optimization

Finding the global minimum, $\psi(y) = \min\{y(\mathbf{x}) : x \in \chi\}$, of a function which is expensive to evaluate is an extensively investigated optimization problem. (Finding the maximum is reformulated as $\min -y(\mathbf{x})$, and the following results apply.) Jones et al. (1998) proposed an efficient sequential solution via the improvement function to assess the gain if a new evaluation is made at $\mathbf{x}$. The improvement function is

$$I(\mathbf{x}) = \max\{y_{\min}^{(n)} - y(\mathbf{x}), 0\},$$

where $y_{\min}^{(n)}$ is the minimum value of y found so far with n runs. The objective is improved by $y_{\min}^{(n)} - y(\mathbf{x})$ if $y_{\min}^{(n)} > y(\mathbf{x})$, otherwise there is no improvement.

The GP statistical model outlined in Section 7.4 leads to a Gaussian predictive distribution for $f(y|\mathbf{x})$, i.e., $y(\mathbf{x}) \sim \mathcal{N}[\hat{y}(\mathbf{x}), s^2(\mathbf{x})]$. The Gaussian predictive model leads to a simple, closed form for the expected improvement,

$$\mathrm{E}\{I(\mathbf{x})\} = s(\mathbf{x})\phi(u) + \{y_{\min}^{(n)} - \hat{y}(\mathbf{x})\}\Phi(u), \tag{7.1}$$

where

$$u = \frac{y_{\min}^{(n)} - \hat{y}(\mathbf{x})}{s(\mathbf{x})},$$

while ϕ and Φ denote the standard Normal probability density function (pdf) and cumulative distribution function (cdf), respectively.

Large values of the first term in (7.1) support global exploration in regions of the input space sparsely sampled so far, where $s(\mathbf{x})$ is large. The second term favors search where $\hat{y}(\mathbf{x})$ is small, which is often close to the location giving $y_{\min}^{(n)}$, i.e., local search. This trade-off between local and global search makes EI-based sequential design very efficient, and it often requires relatively few computer-model evaluations to achieve a desired accuracy in estimating $\min y$.

For instance, in the tidal-power application, the EI surface displayed in Figure 7.1(d) indicates that the first follow-up run is at the location giving the maximum predicted power; see Figure 7.2(a). Thus, the local-search component dominates. Conversely, the suggested location for the second follow-up run is in an unsampled region near the maximum predicted power; see Figure 7.2(c).

Attempts have been made to control this local versus global trade-off for faster convergence (i.e., using as few runs as possible) to the true global minimum. For instance, Schonlau et al. (1998) proposed an exponentiated improvement function, $I^g(\mathbf{x})$, for $g \geq 1$. With $g > 1$, there is more weight on larger improvements when expectation is taken to compute EI. Such large improvements will have a non-trivial probability even if $\hat{y}(\mathbf{x})$ is unfavorable, provided $s(\mathbf{x})$ is sufficiently large. Hence, global exploration of high-uncertainty regions can receive more attention with this adaptation. Similarly, Sóbester et al. (2005) developed a weighted expected improvement function (WEIF) by introducing a user-defined weight parameter $w \in [0, 1]$ in the EI criterion of Jones et al. (1998), and Ponweiser et al. (2008) proposed clustered multiple generalized expected improvement.

7.3.2 EI for Contour Estimation

Ranjan et al. (2008) developed an EI criterion specific to estimating a threshold (or contour) of y. They applied it to a 2-queue 1-server computer network simulator that models the average delay in a queue for service.

Let the feature of interest $\psi(y)$ be the set of input vectors $\mathbf{x}$ defining the contour at level a, viz.

$$S(a) = \{x : y(\mathbf{x}) = a\}. \tag{7.2}$$

The improvement function proposed by Ranjan et al. (2008) is

$$I(\mathbf{x}) = \epsilon^2(\mathbf{x}) - \min[\{y(\mathbf{x}) - a\}^2, \epsilon^2(\mathbf{x})], \tag{7.3}$$

where $\epsilon(\mathbf{x}) = \alpha s(\mathbf{x})$ for a positive constant α (e.g., $\alpha = 1.96$, corresponding to 95% confidence/credibility under approximate Normality). This improvement function defines a limited region of interest around $S(a)$ for further experimentation. Point-wise, the extent of the region depends on the uncertainty $s(\mathbf{x})$ and hence the tolerance $\epsilon(\mathbf{x})$.

Under a Normal predictive distribution, $y(\mathbf{x}) \sim \mathcal{N}[\hat{y}(\mathbf{x}), s^2(\mathbf{x})]$, the expectation of $I(\mathbf{x})$ can again be written in closed form, namely

$$\begin{aligned} \mathrm{E}\{I(\mathbf{x})\} &= [\epsilon^2(\mathbf{x}) - \{\hat{y}(\mathbf{x}) - a\}^2]\,\{\Phi(u_2) - \Phi(u_1)\} \\ &\quad + s^2(\mathbf{x})\,[\{u_2\phi(u_2) - u_1\phi(u_1)\} - \{\Phi(u_2) - \Phi(u_1)\}] \\ &\quad + 2\{\hat{y}(\mathbf{x}) - a\}s(\mathbf{x})\,\{\phi(u_2) - \phi(u_1)\}\,, \end{aligned} \tag{7.4}$$

where

$$u_1 = \frac{a - \hat{y}(\mathbf{x}) - \epsilon(\mathbf{x})}{s(\mathbf{x})} \quad \text{and} \quad u_2 = \frac{a - \hat{y}(\mathbf{x}) + \epsilon(\mathbf{x})}{s(\mathbf{x})}\,.$$

Like EI for optimization, the EI criterion in (7.4) trades off the twin aims of local search near the predicted contour of interest and global exploration. The first term on the right of (7.4) recommends an input location with a large $s(\mathbf{x})$ in the vicinity of the predicted contour. When it dominates, the follow-up point is often essentially the maximizer of $\epsilon^2(\mathbf{x}) - \{\hat{y}(\mathbf{x}) - a\}^2$. This consideration led to the new point in Figure 7.3(c) of the volcano application, for instance. The last term in (7.4) gives weight to points far away from the predicted contour with large uncertainties. The second term is often dominated by the other two terms in the EI criterion.

The EI criterion in (7.4) can easily be extended to related aims. For simultaneous estimation of k contours $S(a_1), \ldots, S(a_k)$, with $S(\cdot)$ defined in (7.2), the improvement function becomes

$$I(\mathbf{x}) = \epsilon^2(\mathbf{x}) - \min\left[\{y(\mathbf{x}) - a_1\}^2, \ldots, \{y(\mathbf{x}) - a_k\}^2, \epsilon^2(\mathbf{x})\right],$$

and the corresponding EI can also be written in a closed form. Roy (2008) considered estimation of the contour defining a given percentile of the output distribution when inputs to the computer model are random (described further in Section 7.5).

7.4 Gaussian Process Models and Predictive Distributions

Evaluation of an EI criterion requires the computation of the expectation of $I(\mathbf{x})$ with respect to the predictive distribution of $y(\mathbf{x})$. In principle, any predictive distribution can be used, but for the method to be useful, it should faithfully reflect the data obtained up to the run in question. In practice, treating the data from the computer-model runs as a realization of a GP is nearly ubiquitous in computer experiments. A GP model leads to a Gaussian predictive distribution, which in turn leads to the closed form expressions in (7.1) and (7.4) and easy interpretation of the trade-off between local and global search.

A GP model is a computationally inexpensive statistical emulator of a computer code. A key feature of many codes is that they are deterministic: rerunning the computer model with the same values for all input variables will give the same output values. Such a deterministic function is placed within a statistical framework by considering a given computer-model input-output relationship as the realization of a stochastic process, $Z(\mathbf{x})$, indexed by the input vector. A single realization of the process is non-random, hence the relevance for a deterministic computer code. For a continuous function, the process is usually assumed to be Gaussian, possibly after transformation, as was done for the volcano application.

This GP or Gaussian Stochastic Process (GaSP) paradigm for modeling a computer code dates back to Sacks et al. (1989a,b), Currin et al. (1991), and O'Hagan (1992). Specifically, the code output function, $y(\mathbf{x})$, is treated as a realization of $Y(\mathbf{x}) = \mu(\mathbf{x}) + Z(\mathbf{x})$, where $\mu(\mathbf{x})$ is a mean (regression) function in $\mathbf{x}$, and $Z(\mathbf{x})$ is a Gaussian process with mean 0 and variance σ^2.

Crucial to this approach is the assumed correlation structure of $Z(\mathbf{x})$. For two configurations of the d-dimensional input vector, $\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{x}' = (x'_1, \ldots, x'_d)$, the correlation between $Z(\mathbf{x})$ and $Z(\mathbf{x}')$ is denoted by $R(\mathbf{x}, \mathbf{x}')$. Here, $R(\cdot, \cdot)$ is usually a parametric family of functions, for which there are many choices (e.g., Santner et al., 2003, Section 2.3). The computations for the applications in Section 7.2 were based on a constant (intercept) regression and a stationary power-exponential correlation function,

$$R(\mathbf{x}, \mathbf{x}') = \exp\left(-\sum_{j=1}^{d} \theta_j |x_j - x'_j|^{p_j} \right).$$

Here, θ_j (with $\theta_j \geq 0$) and p_j (with $1 \leq p_j \leq 2$) control the properties of the effect of input variable j on the output. A larger value of θ_j implies greater sensitivity (activity) of y with respect to x_j, whereas a larger value of p_j implies smoother behavior of y as a function of x_j.

Under this model the output values from n runs of the code, $Y_1, \ldots, Y_n$, have a joint multivariate Normal distribution. If the parameters in the statistical model — in the mean function, in the correlation function, and σ^2 — are treated as known, the predictive distribution of Y at a new $\mathbf{x}$ has a Normal distribution, $\mathcal{N}[\hat{y}(\mathbf{x}), s^2(\mathbf{x})]$, where $\hat{y}(\mathbf{x})$ is the conditional mean of $Y(\mathbf{x})$ given $Y_1, \ldots, Y_n$, and $s^2(\mathbf{x})$ is the conditional variance. Without assuming Normality, $\hat{y}(\mathbf{x})$ can also be interpreted as the best linear unbiased predictor, and $s^2(\mathbf{x})$ is the associated mean squared error. In practice, the unknown parameters have to be estimated, usually by maximum likelihood or Bayesian methods. The predictive distribution is then only approximately Normal, and $\hat{y}(\mathbf{x})$ and $s(\mathbf{x})$ are computed from estimated parameter values. Moreover, Bayesian estimation of the correlation parameters may be necessary to capture all sources of uncertainty in the predictive distribution.

As mentioned already, neither a GP model nor a Normal predictive distribution are essential for sequential design with an EI criterion. For instance,

Chipman et al. (2012) used the optimization improvement function of Jones et al. (1998) with Bayesian additive regression trees (BART). Thus, the emulator was a non-parametric ensemble of tree models.

7.5 Other EI-Based Criteria

Over the last two decades, a plethora of EI-based criteria have been proposed for other scientific and engineering objectives.

Applications can involve several outputs of interest. For instance, constrained optimization problems arise where the code generating the objective function $y(\mathbf{x})$ or another code gives values for a constraint function, $c(\mathbf{x})$, (or several functions). For a feasible solution, $c(\mathbf{x})$ must lie in $[a, b]$. If $c(\mathbf{x})$ is also expensive to compute, one can build an emulator, $\hat{c}(\mathbf{x})$, for it too. The predictive distribution for $c(\mathbf{x})$ leads to an estimate of the probability that $a < c(\mathbf{x}) < b$ for any new run $\mathbf{x}$ under consideration. EI in (7.1) is multiplied by this probability of feasibility to steer the search to locations where EI for the objective $y(\mathbf{x})$ is large and $c(\mathbf{x})$ is likely to be feasible (Schonlau et al., 1998). For a code with multivariate output, Henkenjohann and Kunert (2007) proposed an EI criterion for estimating the global maximum of the desirability scores of simulator outputs.

In a computer experiment, each simulator run requires a fixed, "known" set of values for the $\mathbf{x}$ input vector, but in the real world the input variables may vary randomly, say due to different environmental conditions. Thus, there is an induced output distribution, and its properties can be of interest. Roy (2008) estimated the contour in the input space defining the pth percentile, ν_p, of the output distribution for given $0 < p < 1$. At any iteration of the sequential search based on a limited number of simulator runs, the distribution of the inputs, $\mathbf{x}$, is propagated through the predictive model, giving a distribution of the output variable. In practice, this is achieved by taking a large random sample from the known distribution of the inputs and evaluating $\hat{y}(\mathbf{x})$ for each sampled value. This Monte Carlo sample gives an estimate of the output distribution and hence an estimate $\hat{\nu}_p$ of ν_p. The improvement function for the next point is

$$I^g(\mathbf{x}) = \epsilon^g(\mathbf{x}) - \min[\{y(\mathbf{x}) - \hat{\nu}_p\}^g, \epsilon^g(\mathbf{x})].$$

For $g = 2$ this is the improvement function in (7.3) with $a = \hat{\nu}_p$. Hence, the contour of interest, a, is not fixed throughout the sequential design procedure of Ranjan et al. (2008) but adapts as $\hat{\nu}_p$ changes. Bichon et al. (2009) adapted this criterion to estimate the probability of rare events and system failure in reliability-based design optimization.

Sometimes the simulator input variables partition into those that are controllable in reality and those that are uncontrollable. In a manufacturing pro-

cess context, for instance, these two types of input could be nominal engineering dimensions and random environmental conditions, respectively. Lehman et al. (2004) developed improvement functions for finding M- and V-robust designs for optimization for such a process. For a given configuration of the control variables, $\mathbf{x}_c$, let $\mu(\mathbf{x}_c)$ and $\sigma^2(\mathbf{x}_c)$ be the unknown mean and variance of the simulator output y with respect to the distribution of the uncontrollable variables. An M-robust engineering design minimizes $\mu(\mathbf{x}_c)$ with respect to $\mathbf{x}_c$ subject to a constraint on $\sigma^2(x_c)$, whereas V-robust engineering design minimizes $\sigma^2(x_c)$ subject to a constraint on $\mu(x_c)$.

The inclusion of measurement error (or equivalently, considering a non-deterministic simulator) is becoming more popular in computer experiments, often due to unavoidable simulator biases and inaccurate modeling assumptions. It is undesirable to ignore the measurement error when minimizing a noisy output response, and Ranjan (2013) recommended minimizing a lower quantile, $q(\mathbf{x})$, via an estimate $\hat{q}(\mathbf{x})$ from the predictive distribution, e.g., $\hat{q}(\mathbf{x}) = \hat{y}(\mathbf{x}) - 1.96s(\mathbf{x})$ under a Normal predictive distribution. The proposed improvement function is

$$I(\mathbf{x}) = \max\{0, \hat{q}_{\min}^{(n)} - q(\mathbf{x})\},$$

where $\hat{q}_{\min}^{(n)}$ is the minimum $\hat{q}(\mathbf{x})$ from n runs so far, and $q(\mathbf{x}) = y(\mathbf{x}) - 1.96s(\mathbf{x})$ is an unobservable random quantity. Treating $s(\mathbf{x})$ as non-stochastic and assuming $y(\mathbf{x}) \sim \mathcal{N}[\hat{y}(\mathbf{x}), s^2(\mathbf{x})]$, the corresponding EI criterion is

$$\mathrm{E}\{I(\mathbf{x})\} = s(\mathbf{x})\phi(u) + \{\hat{q}_{\min}^{(n)} - \hat{y}(\mathbf{x}) + 1.96s(\mathbf{x})\}\Phi(u), \tag{7.5}$$

where

$$u = \frac{\hat{q}_{\min}^{(n)} - \hat{y}(\mathbf{x}) + 1.96s(\mathbf{x})}{s(\mathbf{x})}.$$

Like the EI criterion in (7.1), EI in (7.5) facilitates the trade-off between local and global search. One can easily generalize this EI criterion to $\mathrm{E}\{I^g(\mathbf{x})\}$ (as in Schonlau et al., 1998) or introduce a user specified weight (as in Sóbester et al., 2005).

For complex physical phenomena like climate and tidal power, multiple computer simulators with different computational demands are often available for experimentation. For instance, there are 2D and 3D codes for the tidal-power application; the 3D version is a higher-fidelity representation of reality but is much more expensive to run. They can be combined to obtain more informed prediction, and Huang et al. (2006) proposed augmented expected improvement for finding the global minimum of the highest-fidelity process, subject to noise.

7.6 Summary

The essence of these approaches for sequential computer experiments is to formulate the scientific objective through an improvement function. Following some initial runs, the next run is chosen to maximize the expected improvement. In contrast to physical experiments, sequential design is convenient, with the computer handling the logistics of iterating analysis of the data so far, choice of the next run, and making the new run.

With objectives like optimization and contouring in high-dimensional applications, sequential strategies are efficient in terms of solving the problem with a relatively small number of runs. For these reasons, we expect this area of the design of computer experiments will continue to receive considerable research attention from methodologists and users.

Of course, the usefulness of this strategy depends on having a computer model that provides a satisfactory description of the physical process of interest. Such models have to be checked by reference to real data from the physical process (Bayarri et al., 2007). However, once a model has been validated it provides an efficient route to achieving the goals discussed here.

Acknowledgments

The authors thank the editor and an associate editor for their insightful comments leading to clarifications. Funding in partial support of this work was provided by the Canada Research Chairs Program and the Natural Sciences and Engineering Research Council of Canada.

About the Authors

Derek Bingham is a professor of statistics and actuarial science at Simon Fraser University and a Canada Research Chair in industrial statistics. He holds degrees in mathematics and statistics from Concordia, Carleton, and Simon Fraser University. His primary research interests lie in the design and analysis of physical and computer experiments. He was the 2013 recipient of the CRM–SSC Prize, offered jointly by the Centre de recherches mathématiques (in Montréal) and the Statistical Society of Canada.

Pritam Ranjan is an associate professor in the Department of Mathematics and Statistics at Acadia University, Wolfville, Nova Scotia. He obtained BStat and MStat degrees from the Indian Statistical Institute in Kolkata, and a PhD degree in statistics from Simon Fraser University. His research interests include design and analysis of computer experiments, statistical models for computer simulators, sequential designs for feature estimation, and fractional factorial designs with randomization restrictions.

William J. Welch joined the Department of Statistics at the University of British Columbia as a professor in 2003 and was head from 2003 to 2008. Prior to that he was at the University of Waterloo for 16 years. He received his PhD from Imperial College London. His research has concentrated on computer-aided design of experiments, quality improvement, the design and analysis of computer experiments, and drug-discovery from high-throughput screening data. He won the American Statistical Association's Statistics in Chemistry Prize and is an associate director of the Canadian Statistical Sciences Institute.

Bibliography

Bayarri, M. J., Berger, J. O., Calder, E. S., Dalbey, K., Lunagomez, S., Patra, A. K., Pitman, E. B., Spiller, E. T., and Wolpert, R. L. (2009). Using statistical and computer models to quantify volcanic hazards. *Technometrics*, 51:402–413.

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C.-H., and Tu, J. (2007). A framework for validation of computer models. *Technometrics*, 49:138–154.

Bichon, B. J., Mahadevan, S., and Eldred, M. S. (2009). Reliability-based design optimization using efficient global reliability analysis. In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA 2009–2261, Palm Springs, CA. American Institute of Aeronautics and Astronautics.

Chipman, H. A., Ranjan, P., and Wang, W. (2012). Sequential design for computer experiments with a flexible Bayesian additive model. *The Canadian Journal of Statistics*, 40:663–678.

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. *Journal of the American Statistical Association*, 86:953–963.

Henkenjohann, N. and Kunert, J. (2007). An efficient sequential optimization approach based on the multivariate expected improvement criterion. *Quality Engineering*, 19:267–280.

Huang, D., Allen, T. T., Notz, W. I., and Miller, R. A. (2006). Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32:369–382.

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

Lehman, J. S., Santner, T. J., and Notz, W. I. (2004). Designing computer experiments to determine robust control variables. *Statistica Sinica*, 14:571–590.

Loeppky, J. L., Sacks, J., and Welch, W. J. (2009). Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, 51:366–376.

Morris, M. D. and Mitchell, T. J. (1995). Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43:381–402.

O'Hagan, A. (1992). Some Bayesian numerical analysis. In *Bayesian Statistics 4*, pp. 345–363. Oxford University Press.

Ponweiser, W., Wagner, T., and Vincze, M. (2008). Clustered multiple generalized expected improvement: A novel infill sampling criterion for surrogate models. In *2008 IEEE Congress on Evolutionary Computation (CEC 2008)*, pp. 3515–3522.

Ranjan, P. (2013). Comment: EI criteria for noisy computer simulators. *Technometrics*, 55:24–28.

Ranjan, P., Bingham, D., and Michailidis, G. (2008). Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, 50:527–541.

Ranjan, P., Haynes, R., and Karsten, R. (2011). A computationally stable approach to Gaussian process interpolation of deterministic computer simulation data. *Technometrics*, 53:366–378.

Roy, S. (2008). *Sequential-Adaptive Design of Computer Experiments for the Estimation of Percentiles*. Doctoral dissertation, Ohio State University, Columbus, OH.

Sacks, J., Schiller, S. B., and Welch, W. J. (1989a). Designs for computer experiments. *Technometrics*, 31:41–47.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989b). Design and analysis of computer experiments (with discussion). *Statistical Science*, 4:409–435.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. Springer, New York.

Schonlau, M., Welch, W. J., and Jones, D. R. (1998). Global versus local search in constrained optimization of computer models. In *New Developments and Applications in Experimental Design*, volume 34, pp. 11–25. Institute of Mathematical Statistics, Hayward, CA.

Sóbester, A., Leary, S. J., and Keane, A. J. (2005). On the design of optimization strategies based on global response surface approximation models. *Journal of Global Optimization*, 33:31–59.