Confidence Regions for the Location of Response Surface Optima: the R Package OptimaRegion

Enrique del Castillo **Peng Chen** **Adam Meyers** **John Hunt** **James Rapkin**

Abstract

This paper describes the R package **OptimaRegion** for the computation of confidence regions on the location of the optima (global maxima or minima) of response surface models. Both parametric (quadratic and cubic polynomials in up to 5 covariates) and nonparametric models (thin plate splines in 2 covariates) are supported. The methods are based on bootstrapping and Tukey's data depth, and therefore their performance does not rely on any distributional assumption about the response. A coverage analysis is presented demonstrating the quality of the regions found. The package also contains an R implementation of the `Gloptipoly` algorithm for the global optimization of polynomial responses subject to bounds.

Keywords: Nonparametric regression, Response Surface Methodology, Optimization, Data-depth .

Introduction

The goal of many experiments in engineering and science is to find either the maximum, or "peak", or the minimum, or "deepest valley", of some response of interest. How to design and analyze optimization experiments are problems that pertain to the classical field of Response Surface Methodology (RSM) (Box and Draper 1987; Del Castillo 2007). The classical approach in RSM consists in optimizing a fitted model obtained from experimental data, treating it as if it were the true input/output description of the system under study, neglecting the inherent uncertainty of the fitted model. From a frequentist point of view, any property or characteristic of a response surface fitted from experimental data is subject to sampling variability, and hence it should be possible, in principle, to conduct statistical inference on it. Solutions to the problem of statistical inference in RSM have been proposed, usually assuming a polynomial response surface model fitted with ordinary least squares under a normality assumption (Myers and Montgomery 1995; Del Castillo 2007).

One of the most useful inferences in RSM is that of finding a confidence region (CR) on the location of the global maximum or minimum of a response surface. These CRs have found several applications in engineering and science. For instance, Carter, Wampler, Stablein, and Campbell (1982) proposed the idea of using a CR for the optimal dose combination of an anti-Cancer drug as a way to test for therapeutic synergism. If the CR for the optimal dose combination excludes all zero-dose treatment combinations, then there is statistically significant evidence that all of the components are therapeutically synergistic. Otherwise, there are components that can be eliminated from the formulation, a possibility of interest to companies wishing to reduce costs. Also related to pharmaceuticals, a CR on the optima of a response is useful for finding a "design space" in drug development (Peterson 2008). In general, a CR on the optimal settings of a production process is useful in industrial experiments as their size provides a measure of robustness. It also provides a set of solutions within which the engineer can "tweak" the optimal

recipe without jeopardizing the expected system response (Del Castillo 2007). A quite different application comes from evolutionary biology. Brooks, Hunt, Blows, Smith, Bussiere, and Jennions (2005) use a CR on the maxima of experimentally observed fitness responses to test whether an animal has achieved stabilizing selection.

Previous work on CRs for the location of response surface optima assume normal-distributed errors and a quadratic polynomial form (Peterson, Cahya, and del Castillo 2002; Cahya, del Castillo, and Peterson 2004; Wan, Liu, Bretz, and Han 2016), with the first authors providing MATLAB code for up to 3 experimental factors. Early work on confidence regions (Box and Hunter 1954) focused on regions for stationary points of response surfaces, not necessarily on optimum points, and hence are of limited value (Del Castillo and Cahya 2001). We make special emphasis in solving the underlying global optimization problem of a response surface that is not necessarily convex or concave, a problem that has plagued this topic since its inception, the possibility of non-gaussian errors, and the use of either polynomial models of higher order and in higher dimensions or more flexible spline models. In this paper, we discuss and illustrate methods implemented in the R package **OptimaRegion** for the computation and display of distribution-free CRs on the location of global optima of both polynomial and thin plate spline models. The CRs are data-depth based, and follow recent results on the computation of confidence regions of parametric functions using bootstrapping.

Description of the problem

We wish to find a confidence region (CR) for the (global) optima of a function in k variables fitted from observed experimental data without relying in multivariate normality or any other distributional assumption of the data. We assume in this paper a maximization goal without loss of generality. In this paper, bootstrapping methods and their software implementation are presented that provide *valid* and *unbiased* confidence regions for the optima of a function fitted either using a linear regression (polynomial) model or a thin plate spline model. A **valid** $1-\alpha$ CR for a parameter θ, $C^{\theta}_{1-\alpha}$, is a set such that $P(\theta \in C^{\theta}_{1-\alpha}) \geq 1-\alpha$. Interest is of course in CR's that are smallest in size and still have confidence level of at least $1-\alpha$, and hence we will consider not only the coverage but the area of the CR's. Also, a $1-\alpha$ CR is **unbiased** if $P(\theta' \in C^{\theta}_{1-\alpha}) \leq 1-\alpha$ for all $\theta' \neq \theta$ (Casella and Berger 2002). That is, the probability of covering any wrong parameter should always be less than the probability of covering the true parameter.

More specifically, we wish to find a CR for the function:

$$\mathbf{x}^* \;=\; h(\mathbf{x};\hat{\boldsymbol{\beta}}) \;=\; \arg\max f(\mathbf{x},\hat{\boldsymbol{\beta}})$$

where $f(\mathbf{x},\hat{\boldsymbol{\beta}})$ is *either* a polynomial regression model in $\mathbf{x}$ *or* a Thin Plate Spline model in $\mathbf{x}$. In both cases, we assume maximization without loss of generality, and subject the optimum to lie in a region defined by linear bounds on the regressors. In the polynomial model case, $\mathbf{x}^* \in \mathbb{R}^k$ is a random vector with a sampling distribution that depends on the sampling distribution of the $p \times 1$ least squares estimator $\hat{\boldsymbol{\beta}}$ in $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ where $\boldsymbol{X}$ is a $n \times p$ design matrix with columns corresponding to the terms in the quadratic polynomial model $f(\mathbf{x},\hat{\boldsymbol{\beta}})$, and the random errors ε_i in $\boldsymbol{\varepsilon}$ are to be i.i.d. with zero mean, constant variance and with an unknown and unspecified distribution.

Direct bootstrapping approach

A direct application of the idea of bootstrapping consists in fitting many response surface models, globally optimizing each and trimming the outmost α percent $\mathbf{x}^* = h(\boldsymbol{\beta})$ vectors using some

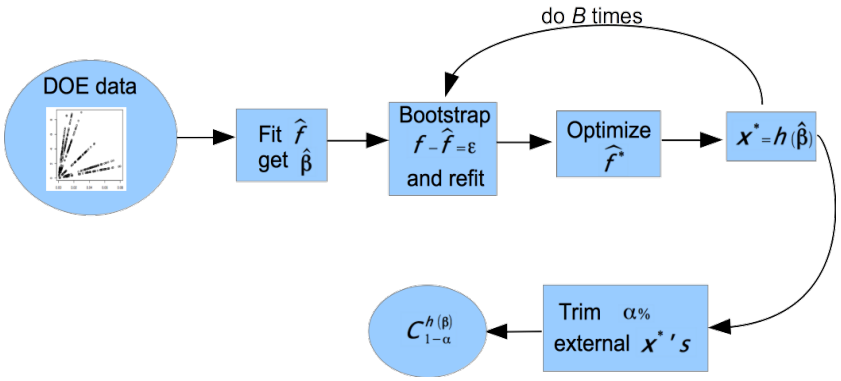

Figure 1: A direct bootstrapping approach to compute the CR on the optima of a response surface $f(x, \boldsymbol{\beta})$, $C^{h(\beta)}_{1-\alpha}$ fitted to experimental data.

method that orders interior and exterior multivariate data (see Figure 1). Unfortunately (see Table 1 below), this method does not provide valid confidence regions, i.e., the coverage provided is smaller than the advertised coverage. The reason, as discussed by Woutersen and Ham (2013), is that by trimming the $h(\boldsymbol{\beta})$ values we are eliminating extreme observations of h that occurred because either a) $\boldsymbol{\beta}$ was very extreme or b) because $\boldsymbol{\beta}$ is not very extreme but $h(\boldsymbol{\beta})$ is extreme. A CR on $h(\boldsymbol{\beta})$ should exclude instances where h, and not $\boldsymbol{\beta}$, are extreme. This is achieved with the method implemented in the **OptimaRegion** package, based on Rao's confidence region approach (Rao 1973) but not with the direct approach. As mentioned by Wan *et al.* (2016), Rao's projection method is the only confidence set available that guarantees the $(1-\alpha)$ confidence level and hence it is the basis of our bootstrapping method.

Implementation of the bootstrapping methods in OptimaRegion

OptimaRegion implements a bootstrapping approach for confidence regions of response surface optima based on Rao (1973) (p. 473) projection idea, mapping the confidence set of response parameters to the confidence set of optima in a discrete or pointwise manner.

1 Obtain a $100(1-\alpha)\%$ CR for $\boldsymbol{\beta}$, $C^{\boldsymbol{\beta}}_{1-\alpha}$, from the asymptotic distribution of $\hat{\boldsymbol{\beta}}$.

2 For each $\boldsymbol{\beta} \in C^{\boldsymbol{\beta}}_{1-\alpha}$, evaluate $h(\boldsymbol{\beta})$.

3 Let $C^{h(\boldsymbol{\beta})}_{1-\alpha} = \{\tau \in \mathbb{R}^k | \tau = h(\boldsymbol{\beta}) \text{ for all } \boldsymbol{\beta} \in C^{\boldsymbol{\beta}}_{1-\alpha}\}$

To estimate this confidence region, we use bootstrapping in steps 1 and 3:

1_B Obtain an estimate of the $100(1-\alpha)\%$ CR for $\boldsymbol{\beta}$ by bootstrapping B instances of $\hat{\boldsymbol{\beta}}$. These instances make $\widehat{C}^{\boldsymbol{\beta}}_{1-\alpha}$;

2_B For each $\boldsymbol{\beta} \in \widehat{C}^{\boldsymbol{\beta}}_{1-\alpha}$, evaluate $h(\boldsymbol{\beta})$.

3_B Let $\widehat{C}^{h(\boldsymbol{\beta})}_{1-\alpha} = \{\tau \in \mathbb{R}^k | \tau = h(\boldsymbol{\beta}) \text{ for all } \boldsymbol{\beta} \in \widehat{C}^{\boldsymbol{\beta}}_{1-\alpha}\}$

Note that in order to implement this method for $h(\mathbf{x}; \hat{\boldsymbol{\beta}}) = \arg\max f(\mathbf{x}, \hat{\boldsymbol{\beta}})$, we need a means to define the "innermost" $\boldsymbol{\beta}$ parameters in step **1_B**, and an optimization method that finds the global maximums of each $h(\boldsymbol{\beta})$ in step **2_B**. In step **1_B**, **OptimaRegion** uses Tukey's data depth

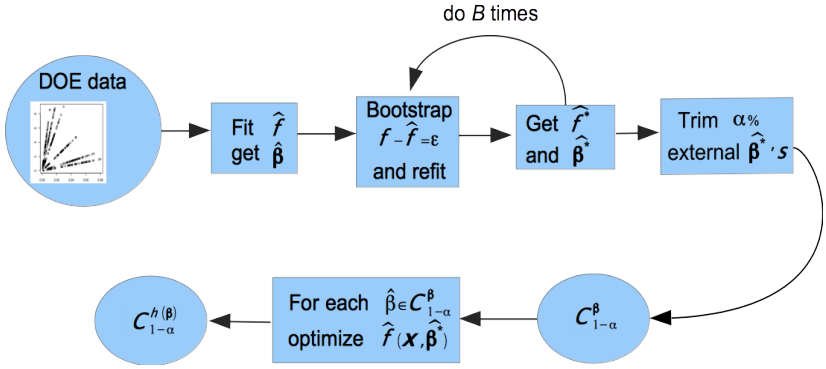

Figure 2: Overview of the projection bootstrapping approach implemented in **OptimaRegion** for finding a CR on the global optima of a function $f(x, \boldsymbol{\beta})$, $C^{h(\beta)}_{1-\alpha}$ from experimental data. First a CR on the response parameters, $C^{\beta}_{1-\alpha}$ is obtained.

(Tukey 1975) as implemented in the **DepthProc** package, and in step $\mathbf{2_B}$ it uses either a nonlinear programming algorithm with multiple restarts as implemented in the **nloptr** package (for thin plate spline and quadratic polynomials subject to linear constraints in two regressors) or our R implementation of the GloptiPoly global optimization algorithm (for higher polynomial models subject to bounds). Tukey's data depth is used to order the B instances $\widehat{\boldsymbol{\beta}}$ and trim the α % outermost (the α % with lowest D_T value; for instance, points such that $D_T(\mathbf{x}) = 0$ define the convex hull of F). This yields $\widehat{C}^{\boldsymbol{\beta}}_{1-\alpha}$ in step $\mathbf{1_B}$.

Furthermore, since we are computing $\widehat{C}^{\boldsymbol{\beta}}_{1-\alpha}$ pointwise for a finite number of B vectors $\boldsymbol{\beta}$, our final confidence region for $h(\boldsymbol{\beta})$ will also be a pointwise region. This means that to end up with a *region* we need some additional rule that defines the boundary of the region. Woutersen and Ham (2013) the authors propose to use an arbitrary quantity $\eta > 0$ and define the CR for $h(\boldsymbol{\beta})$ to be the set of all $\boldsymbol{\beta}$ that are no farther than the euclidean distance η from each of the B $h(\boldsymbol{\beta})$ values (in $\mathbb{R}^2$ the CR will then be composed of the union of B circles around each $\mathbf{x}^*$). While this step was specified in order to be able to proof the validity of the resulting CR, in practice it is not clear how to select the radius η to make the resulting CR as small as possible and avoid overly conservative CR's. **OptimaRegion** displays the CRs by plotting the convex hull of all the points generated. The coordinates of all the generated points inside the CR are returned, and the average or centroid estimate of the optimal points $\mathbf{x}^*$, a "bagging" (bootstrapped aggregated) estimate, is also plotted.

In what follows we concentrate in computational methods for obtaining CR's for $h(\mathbf{x}; \hat{\boldsymbol{\beta}}) = \arg\max f(\mathbf{x}, \hat{\boldsymbol{\beta}})$ subject to linear bounds. The underlying global optimization process of a non-convex function makes finding the desired confidence regions a very difficult problem for $k > 2$. Therefore, **OptimaRegion** contains separate functions for $k = 2$ where a nonlinear programming method is called from a lattice of initial points, and for polynomial functions with $2 < k \leq 5$ using the GloptiPoly global optimization method which guarantees finding the location of the global optima in polynomial models. Both ordinary least-squares-fitted polynomial models and regularized thin plate spline models are considered. The package assumes these models are to be fit from experimental data, so the user only need to enter the response (y) and covariate (x) data.

Types of response surface models considered in package OptimaRegion

The package finds the CR of optima of two classes of response surface models:

- Polynomial response surface models.- These are linear parametric models of the form:

$$y = f(\mathbf{x}, \boldsymbol{\beta}) + \varepsilon$$

where $f(\mathbf{x}, \boldsymbol{\beta}) = \sum_{j=0}^{p-1} \beta_j \phi_j(\mathbf{x})$ where $\phi_j(\mathbf{x}) : \mathbb{R}^{\mathbf{k}} \to \mathbb{R}$ is a polynomial basis, typically with $\phi_0(\mathbf{x}) = 1$. Most common in response surface methodology is the quadratic polynomial case in k numeric factors, where $p = k(k-1)/2 + 2k + 1$ (Myers and Montgomery 1995). Also common in mixture experiments are cubic polynomial models. **OptimaRegion** fits and determines the CR of the optima of up to cubic polynomials in up to $k = 5$ experimental factors. The fitting is based on ordinary least squares using the `lm` function in package **stats**. Bootstrapping is based on the ordinary residuals $y_i - \hat{f}(\mathbf{x}_i)$.

- Thin Plate Spline models.- Low order polynomials as in used in Response Surface methodology are not usually flexible enough to model widely variable functions over a larger experimental region. For an instance in evolutionary biology, Rapkin, Jensen, Archer, House, Sakaluk, Del Castillo, and Hunt (2018) used as a more flexible alternative Thin Plate Splines (TSP) to model trade-offs between immune response and reproductive effort in nutrition experiments with crickets, and computed CRs on the maxima of the fitted functions. TSPs are nonparametric models that have the form $y = f(\mathbf{x}) + \varepsilon$ where $f(\mathbf{x})$ is fitted by solving the penalized sum of squares:

$$\hat{f} = \arg\min_{f \in \mathcal{H}} S(f) = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2 + \lambda J_m(f)$$

where $\lambda > 0$ is a quantity that penalizes the total variation of f, $J_m(f)$. The minimization is done over a Hilbert space $\mathcal{H}$ decomposed as $\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1$ where $\mathcal{H}_0 = \text{span}\{\phi_i(\mathbf{x})\}_{i=1}^{p}$ is the so-called null space of functions $\phi(\mathbf{x})$ that are not penalized and $\mathcal{H}_1$ is a reproducible kernel Hilbert space of functions whose smoothness determines the smoothness of the fitted function, which results in $\hat{f}(\mathbf{x})$ being equal to the sum of specific instances of these two types of functions. Remarkably, the Kimeldorf-Wahba representer theorem shows how the solution to this infinite dimensional optimization problem is given by a finite number of parameters:

$$\hat{\mathbf{f}} = T\hat{\boldsymbol{\beta}} + K\hat{\boldsymbol{\delta}}$$

where T is an $n \times p$ matrix of polynomial functions of order m and K is an $n \times n$ matrix, $K = \{R(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j=1}^{n}$, where the $R(\mathbf{x}_i, \mathbf{x}_j)$ are radial basis functions that depend on the distance r only, i.e., $R(r) = a||r||^{2m-d} \log(r)$ if d is even and $R(r) = a||r||^{rd-m}$ if d is odd (a is a constant that depends on m and d). The vectors of parameters β and δ are obtained from minimizing $S(f)$ (for a proof and details, see Wahba (1990)). This formulation shows the relation between a thin plate spline model and an universal kriging model with a parameter-free spatial covariance model (Nychka 2000). Thus, despite being a nonparametric model, the prediction at a new point $\mathbf{x}_0$ is given by the parametric expression

$$\hat{f}(\boldsymbol{x}_0) = \mathbf{T}_0'\hat{\boldsymbol{\beta}} + \mathbf{K}_0'\hat{\boldsymbol{\delta}}$$

where $\mathbf{T}_0$ and $\mathbf{K}_0$ are $1 \times p$ and $\mathbf{K}_0$ are $1 \times n$ vectors analogous to matrices T and K. Bootstrapping is based on the residuals $\mathbf{y} - T\hat{\boldsymbol{\beta}} - K\hat{\boldsymbol{\delta}}$. **OptimaRegion** fits and computes residuals for this model using the **fields** package (Douglas Nychka, Reinhard Furrer, John Paige, and Stephan Sain 2017). The aforementioned projection method is then based on building first a bootstrapped joint CR on parameters $(\boldsymbol{\beta}, \boldsymbol{\delta})$.

Functions in package OptimaRegion

There are 5 main functions in the **OptimaRegion** package:

Function	Objective
`OptRegionQuad`	Computes distribution-free bootstrapped confidence regions for the location of the optima of a quadratic polynomial model in 2 regressors
`OptRegionTps`	Computes distribution-free bootstrapped confidence regions for the location of the optima of a Thin Plate Spline model in 2 regressors
`GloptiPolyRegion`	Computes distribution-free bootstrapped CRs for the location of global optima for polynomial models up to cubic order in up to 5 regressors
`CRcompare`	Computes bootstrapped confidence intervals for the distance between the optima of two different response surface models, either quadratic polynomials or thin plate spline models
`GloptipolyR`	R implementation of the "Gloptipoly" algorithm (Lasserre 2001) for global optimization of polynomial equations subject to bounds

Examples

Example 1. CR on the maximum of a fitted quadratic polynomial using `OptRegionQuad`.- Consider a mixture-amount experiment in two components (Drug dataset) where the effectiveness of the drug (a percentage) is the response, which in many cases has value zero. Hence, the data cannot be considered normal and classic approaches to find a CR cannot be used. Thus, we try using `OptRegionQuad` as it does not rely on any normality assumption. Given the shape of the experimental region, the `triangularRegion` switch is set to on, with upper and right vertices as specified for `vertex1` and `vertex2` (the other vertex is the origin). This indicates the limits of the experimental region, and therefore, the region where the maxima of the response surface should be sought. Peterson and Novick (2007) pointed out how mixture-amount experiments in 2 components have such a triangular region of interest. The R command is:

```
out <- OptRegionQuad(X = Drug[,1:2], y = Drug[3], nosim = 500, LB = c(0,0),
UB = c(0.08,11), xlab = "Component 1 (mg.)", ylab = "Component 2 (mg.)",
triangularRegion = TRUE, vertex1 = c(0.02,11), vertex2 = c(0.08,1.8),
outputPDFFile = "Mixture_plot.pdf")
```

The resulting 95% confidence region generated in the PDF file is shown in Figure 3, which also shows smoothed contours of the response. Note these are *not* the quadratic polynomial contours. Also, note how the CR is "pushed" against the constraint and results in a "thin line". The red dot is the centroid of all the generated maxima, the bagging estimate of $\mathbf{x}^*$.

Example 2. CR on the global maximum of a fitted Thin Plate Spline model for a mixture-amount experiment using `OptRegionTps`.- Consider next the same mixture-amount experiments as before (`drugs` dataset) but suppose we think the quadratic polynomial model provides is not flexible enough to represent the true surface. Instead, we can try fitting and optimizing a Thin Plate Spline (TPS) model using function `OptRegionTps`.

```
out <- OptRegionTps(X = Drug[,1:2], y = Drug[,3], nosim = 500, lambda = 0.05,
LB = c(0,0), UB = c(0.08,11), xlab = "Component 1 (mg.)", ylab = "Component 2 (mg.)",
triangularRegion = TRUE, vertex1 = c(0.02,11), vertex2 = c(0.08,1.8),
outputPDFFile = "Mixture_plot.pdf")
```

In contrast with example 1, `OptRegionTps` will take a few minutes to complete the computations in a fast PC. Note the parameter `lambda=0.05`; this is the penalization parameter when fitting

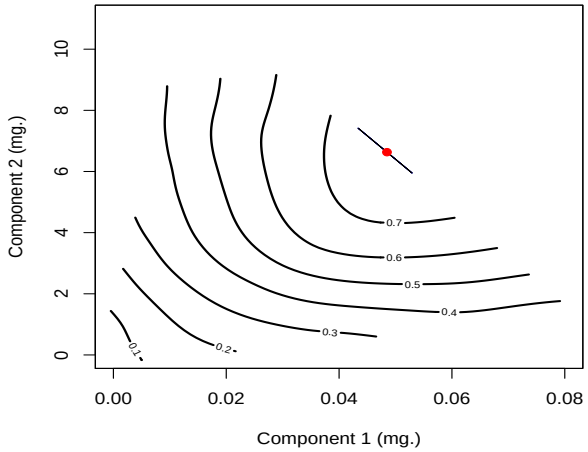

Figure 3: Example 1: a 95% CR on the maximum of a 2-drug mixture amount experiment, `Drugs` datafile. Plot generated with the `OptRegionQuad` function.

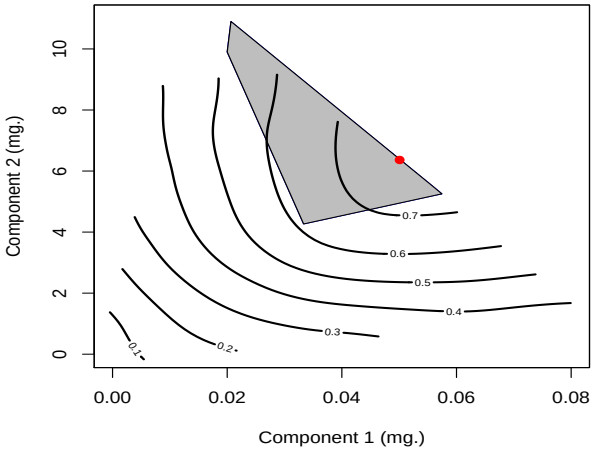

Figure 4: Example 2: a 95% CR on the maximum of a 2-drug mixture amount experiment, `Drugs` datafile. Plot generated with the `OptRegionTps` function.

a TPS model. Larger values of `lambda` make the fitted model less "wiggly". The confidence levels obtained are conditional on the pre-selected value of `lambda` which can be obtained via crossvalidation using the package **fields**. The PDF output file showing the CR plot is shown in Figure 4. In this case, the CR contains area in the interior of the triangular experimental region. The linear boundaries of the shaded CR are the result of using the convex hull of the optima generated by the bootstrapping algorithm. Increasing the number of bootstraps may smooth the boundaries somewhat (i.e., shorter linear segments) but the computation time will increase accordingly. Despite being a better model for this dataset, the more flexible character of the TPS model contains a good deal of uncertainty about the location of the maximum drug components that maximizes the efficacy.

Example 3. CR on the global maximum of a fitted Thin Plate Spline model for a factorial experiment using `OptRegionTps`. We now illustrate the use of the `OptRegionTps` function for an experiment where the factors are centered around zero and the experimental region is a square. Suppose we generate some dummy 'X' and 'y' data by means of Monte Carlo simulation:

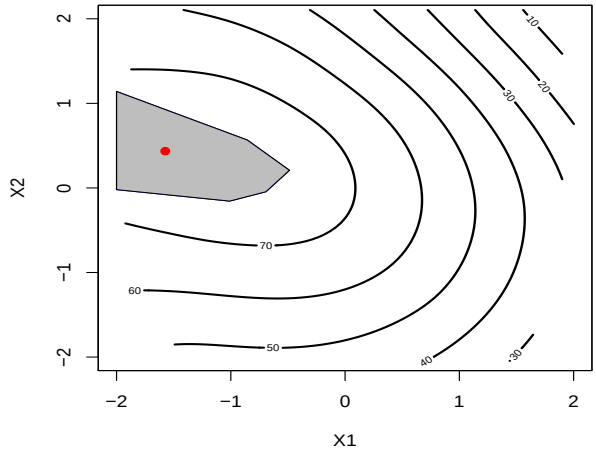

Figure 5: Example 2: a 95% CR on the maximum of a 2 factor randomly generated factorial experiment over a squared region. Plot generated with the `OptRegionTps` function.

```
X <- cbind(runif(100,-2,2), runif(100,-2,2))

y <- as.matrix(72 - 11.78*X[,1] + 0.74*X[,2] - 7.25*X[,1]^2 - 7.55*X[,2]^2 -
  4.85*X[,1]*X[,2] + rnorm(100,0,8))
```

Next we compute a 95% CR on the maxima of a fitted TPS model:

```
out <- OptRegionTps(X = X, y = y, nosim = 200, LB = c(-2,-2), UB = c(2,2),
  xlab = "X1", ylab = "X2")
```

Note we did not specify a triangular region. The PDF file created on completion is shown in Figure 5 and displays the corresponding region, together with the contours of the fitted TPS model.

Example 4. Computing confidence *intervals* on the distance between two response surfaces using `CRcompare`.- Suppose we have experimental data from which we can fit a quadratic polynomial model to each of two different responses. We now wish to investigate if the "peaks" of each response are significantly close. A confidence interval on the distance between the two maxima can be computing with the `CRcompare` function. To use this function, we need to provide the 'X' and 'y' experimental data for each response. We first generate some data for illustration purposes:

```
X1 <- cbind(runif(100,-2,2), runif(100,-2,2))

y1 <- as.matrix(72 - 11.78*X1[,1] + 0.74*X1[,2] - 7.25*X1[,1]^2 - 7.55*X1[,2]^2 -
    4.85*X1[,1]*X1[,2] + rnorm(100,0,8))

X2 <- cbind(runif(100,-2,2), runif(100,-2,2))

y2 <- as.matrix(72 - 11.78*X2[,1] + 0.74*X2[,2] - 7.25*X2[,1]^2 - 7.55*X2[,2]^2 -
    4.85*X2[,1]*X2[,2] + rnorm(100,0,8))
```

We next run the `CRcompare` routine with this input-output data:

```
out <- CRcompare(X1 = X1, y1 = y1, X2 = X2, y2 = y2, responseType = 'Quad',
   nosim1and2 = 200, alpha = 0.05, LB1 = c(-2,-2), UB1 = c(2,2), LB2 = c(-2,-2),
   UB2 = c(2,2) )
```

Note we specified a quadratic ('Quad') response model for both responses and 200 bootstrap iterations. Also note that the lower and upper bounds within which each response may have its maximum can differ ('maximization' is TRUE by default). `CRcompare` will run either `OptRegionQuad` or `OptRegionTps` for each response and compute all the pairwise distances from the two CR's. It will then bootstrap the distances and will output the corresponding bootstrap confidence interval on the mean and median distance:

```
> out$mean
[1] 0.3643884

> out$median
[1] 0.305715

> out$ciMean
     conf
[1,] 0.95 36.43 984.66 0.3324372 0.406087

> out$ciMedian
     conf
[1,] 0.95 18 966.76 0.2833316 0.3490922
```

Hence, a 95% confidence interval on the mean distance is (0.3324,0.4060) and a 95% confidence interval on the median distance is (0.2833,0.3490).

Example 5. Computing a CR on the global optimum of a polynomial model in 3 factors using `GloptiPolyRegion`. Box and Draper (1987) (p. 305) analyze a 3-factor experiment using classical response surface techniques, in particular, canonical analysis, to determine the nature of the optimum of a quadratic polynomial model. In this experiment , the goal is to find the percentage concentration of two constituents (x_1 and x_2, in coded units) and the temperature (x_3, coded) that maximize the elasticity of certain polymer (y). The data for this experiment can be found in file `quad_3D`:

```
> str(quad_3D)
Classes ‘tbl_df’, ‘tbl’ and 'data.frame': 16 obs. of  4 variables:
 $ x1: num  -1 1 -1 1 -1 1 -1 1 -2 2 ...
 $ x2: num  -1 -1 1 1 -1 -1 1 1 0 0 ...
 $ x3: num  -1 -1 -1 -1 1 1 1 1 0 0 ...
 $ y : num  25.7 49 42.8 35.9 41.5 ...
```

The second order polynomial model fitted using ordinary least squares is:

$$\begin{aligned}\hat{f}(x) = \; & 57.31 + 1.5x_1 - 2.13x_2 + 1.81x_3 \\ & - 7.13x_1x_2 - 3.27x_1x_3 - 2.73x_2x_3 - 4.69x_1^2 - 6.27x_2^2 - 5.21x_3^2,\end{aligned}$$

and has an excellent fit with all terms statistically significant, no lack of fit, $R^2 = 0.972$, and all usual diagnostics look adequate (see Del Castillo (2007), chapter 7). The fitted quadratic polynomial has a maximum at $(0.4603, -0.4644, 0.1509)$. At the end of their analysis, Box and Draper (1987) indicated how

> "A more accurate picture about what is known at this stage of experimentation can be gained by the confidence region calculation described in Box and Hunter, 1954"

but provided not such calculation. Del Castillo and Cahya (2001) reanalyzed this experiment, showing how a confidence region on stationary points, such as Box and Hunter's, results in a disjoint region, since, despite the excellent fit, there is a non-neglible probability the function is in reality a saddle function, whose stationary point is located far away from the region where the other stationary point, which corresponds to a maximum, occurs (Box and Hunter regions are CR on *all* the stationary points, not on the global optima. In most engineering and science applications, a CR on true optima is desired). To generate a 90% confidence region for the global maximum, we run the R command:

```
out <- GloptiPolyRegion(
  X = quad_3D[, 1:3], y = quad_3D[, 4], degree = 2,
  lb = c(-2, -2, -2), ub = c(2, 2, 2), B = 500, alpha = 0.1,
  maximization = TRUE, outputPDFFile = "CR_quad_3D.pdf", verbose = TRUE
)
```

and obtain the confidence region in Figure 6. Contrary to the CR on all stationary points, the region obtained is not disjoint and corresponds only to points of maximum response. To determine the resulting set of bootstrapped maxima and the bagging estimate of the global maximum, enter:

```
> str(out)
List of 2
 $ boot_optima  : num [1:1800, 1:3] 0.396 0.561 0.406 0.926 0.29 ...
 $ boost_optimum: num [1:3] 0.502 -0.499 0.149
```

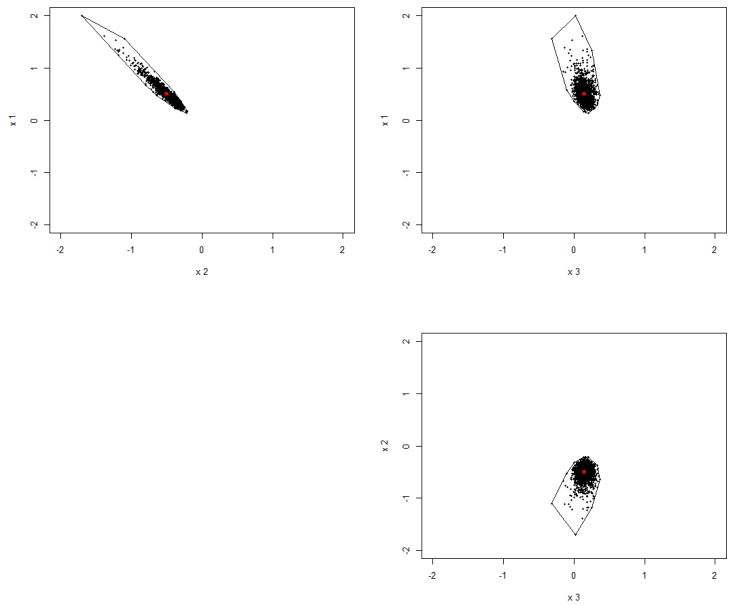

Figure 6: Pairwise projections of a 90% CR for the location of the global maximum in the example in Box and Draper (1987) obtained using `GloptiPolyRegion`.

Example 6. A CR on the global optimum of a 5-factor cubic polynomial model

using `GloptiPolyRegion`. Consider next the function:

$$f(x) = \mu(x) + \epsilon = 10 - (x_1 - 1.5)^2 - (x_2 - 2)^2 - (x_3 - 2.5)^2 - (x_4 - 3)^2 - (x_5 - 3.5)^2$$
$$+0.1x_1^3 - 0.1x_2^3 - 0.1x_3^3 - 0.1x_4^3 - 0.1x_5^3 + x_2x_4 - x_3x_4 + \epsilon$$

with $\epsilon \sim N(0, \sigma^2 = 2^2)$ (i.i.d), defined in the region $R = \{0 \leq x_i \leq 5 \text{ for } i = 1, \cdots 5\}$. The mean response has a global maximum at $x^* = (2.28, 2.44, 1.02, 2.65, 2.54)$. 300 simulated realizations of this function at points $\mathbf{x} \in R$ generated using a Latin Hypercube can be found in the dataset `cubic_5D`. To obtain a 95% CR on the location of the maximum response, we enter the command:

```
out <- GloptiPolyRegion(
  X = cubic_5D$design_matrix, y = cubic_5D$response, degree = 3,
  lb = rep(0, 5), ub = rep(5, 5), B = 200, alpha = 0.05,
  maximization = TRUE, outputPDFFile = "CR_cubic_5D.pdf", verbose = TRUE
)
```

Figure 7 shows a matrix scatter plot generated by this command, displaying the desired CR. To determine the resulting set of bootstrapped maxima and the bagging estimate of the global maximum, enter:

```
> str(out)
List of 2
 $ boot_optima   : num [1:1900, 1:5] 5 2.37 2.44 2.89 5 ...
 $ bagged_optimum: num [1:5] 3.85 2.38 1 2.62 2.49
```

Numerical evaluation of coverage probability

For a given point $\mathbf{x}$ (equal to $\mathbf{x}^*$ or any other point), the coverage is defined as the proportion of times $\mathbf{x} \in \widehat{C}^{\boldsymbol{\beta}}_{1-\alpha}$ in N_s trials from simulated data. Wei and Lee (Wei and Lee 2012) show how a data-depth confidence region is second order accurate, that is, its coverage error (the difference between the actual coverage and the nominal confidence level) is of order n^{-1} where n denotes the sample size. They showed this result holds for different depth measures, including Tukey's data depth measure. Here we evaluate the performance of the functions `OptRegionQuad`, `GloptiPolyRegion` and `OptRegionTps` in package **OptimaRegion** via Monte Carlo simulation, focusing on the coverage and size of the resulting confidence regions.

1) Coverage of CR on the optima of a quadratic polynomial model in two covariates–general nonlinear optimization.

A CR for the optima of a quadratic polynomial model using the method described above is obtained using the `OptRegionQuad` function. Table 1 shows some coverage levels for the global maximum of the simulated response surface $f(\mathbf{x})$ compared with the direct bootstrapping approach referred earlier. Here

$$f(\mathbf{x}) = 90.79 - 1.095x_1 - 1.045x_2 - 0.775x_1x_2 - 2.781x_1^2 - 2.524x_2^2$$

to which i.i.d. $N(0, \sigma^2)$ noise was added. This function has a single maximum at $\mathbf{x}^* = (-0.1716, -0.1806)'$. The points $\mathbf{x}$ at which the function was simulated were the 11 runs in a rotatable Central Composite Design with a domain of radius $\sqrt{2}$ around the origin (Box and

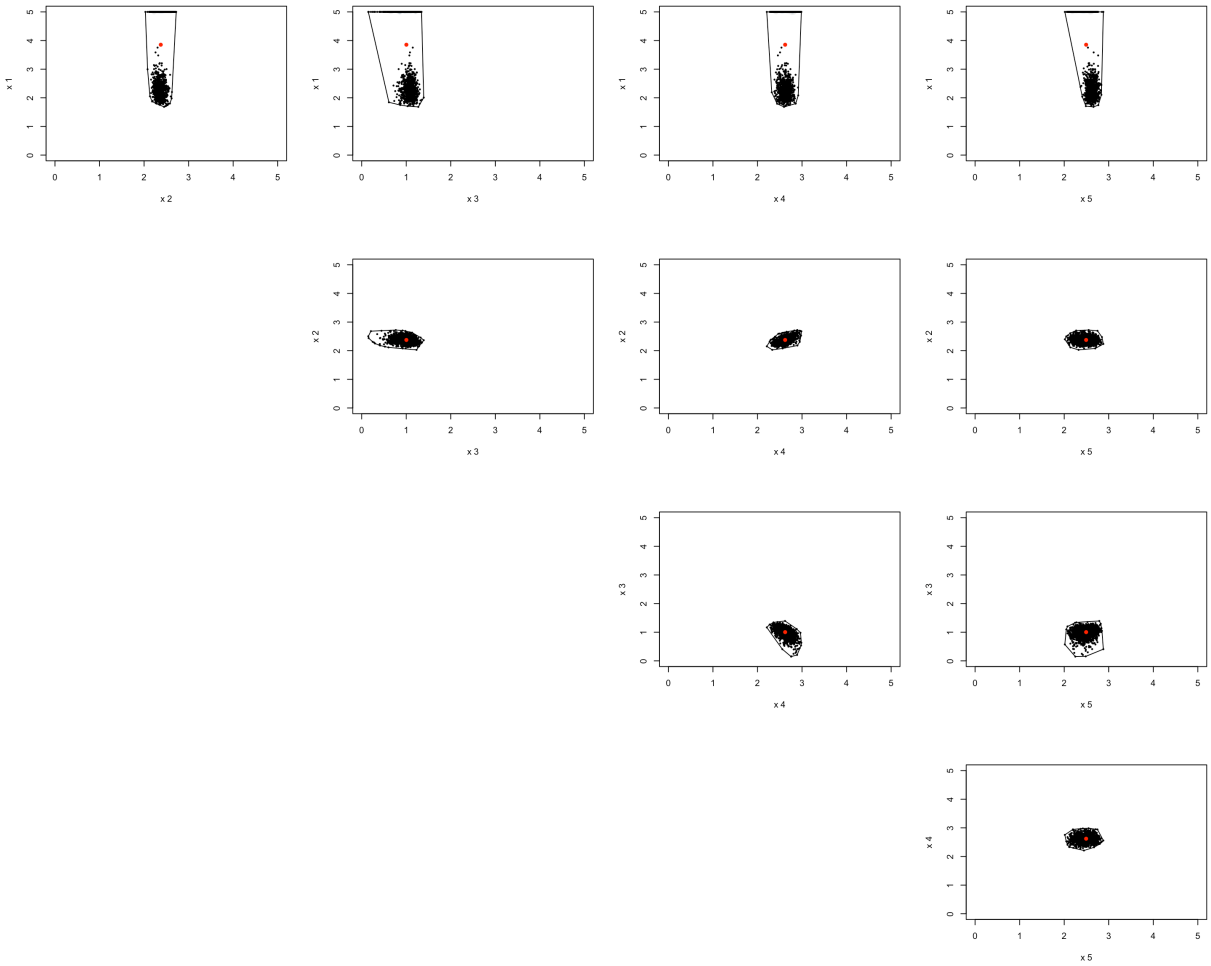

Figure 7: Pairwise projections of a 95% CR on the maximum of $f(x) = 10 - (x_1 - 1.5)^2 - (x_2 - 2)^2 - (x_3 - 2.5)^2 - (x_4 - 3)^2 - (x_5 - 3.5)^2 + 0.1x_1^3 - 0.1x_2^3 - 0.1x_3^3 - 0.1x_4^3 - 0.1x_5^3 + x_2x_4 - x_3x_4$.

Draper 1987; Del Castillo 2007) with 11 runs, in addition to sets of 11 runs randomly generated according to a uniform distribution on the square that goes from $(-\sqrt{2}, -\sqrt{2})$ in its lower left corner to $(\sqrt{2}, \sqrt{2})$ in the upper right corner, giving a total of n observations.

The results on Table 1 show how, compared with the naive bootstrapping approach, only the approach implemented in the `OptRegionQuad` function generates valid confidence regions, although always achieving higher than advertised coverages[1]. A reason for this behavior is that the final CR contour is obtained from the convex hull of the optima $\mathbf{x}^*$ which will tend to provide conservative coverage regardless of n and σ (see Table 2). The direct bootstrap method, in contrast, does not achieve the nominal coverage and cannot be recommended (and hence the OptimaRegion package does not implement it).

However, the areas of the CRs computed by `OptRegionQuad` are quite small, rapidly decreasing in size as n increases (Table 2), a very desirable property. Finally, Table 3 shows that the CRs obtained by `OptRegionQuad` are *unbiased*, since the coverage of non-optimal points is always lower than $1 - \alpha$, with lower coverages the farther is the non-optimal point from $\mathbf{x}^*$.

[1]Recall that the estimated standard error of the estimated coverage $\hat{p}$ is given by $\sqrt{\hat{p}(1-\hat{p})/n}$, so all the estimated coverages presented in this paper are very precise.

CR type	N_s	B	α	n (reps.)	coverage
Direct	1000	1000	0.10	55 (5)	0.843
Direct	1000	1000	0.10	1100 (100)	0.868
`OptRegionQuad`	1000	1000	0.05	1100 (100)	0.981
`OptRegionQuad`	1000	1000	0.10	1100 (100)	0.979
`OptRegionQuad`	1000	1000	0.20	1100 (100)	0.930

Table 1: Estimated coverages of bootstrapped $(1-\alpha)100\%$ CRs for the maximum of a quadratic polynomial regression model. N_s is the number of simulations, B is the number of bootstrapped samples, n is the sample size. Simulated noise has $\sigma = 2$.

$n(reps.)$	coverage	σ	$\overline{\text{area}}$	sd.(area)	$\frac{\overline{\text{area}}}{\max \text{area}}$	$\frac{\text{sd(area)}}{\max \text{area}}$
1100(100)	0.981	2	0.007	0.00087	0.00088	0.0001
2200(200)	0.978	2	0.0036	0.00036	0.00045	0.000045
5500(500)	0.987	2	0.0014	0.00013	0.00018	0.000016
1100(100)	0.988	5	0.052	0.012	0.0065	0.0015
2200(200)	0.984	5	0.023	0.0037	0.0029	0.00046
5500(500)	0.985	5	0.009	0.0011	0.0012	0.00014
1100(100)	0.983	10	0.475	0.4004	0.059	0.0501
2200(200)	0.981	10	0.137	0.068	0.0172	0.0085
5500(500)	0.987	10	0.041	0.0083	0.0052	0.00104

Table 2: Estimated coverages of the **optimal point** of a 95% bootstrapped CR as obtained by `OptRegionQuad` for the maximum of a quadratic polynomial regression model. In all cases, $N_s = 1000$, and $B = 1000$ were used. Maximum area in the search region is $8 = (-\sqrt{2}, \sqrt{2}) \times (-\sqrt{2}, \sqrt{2})$.

2) Coverage of CRs for optima of higher order polynomial models in higher number of variables using the GloptiPoly algorithm

Since the fitted polynomial models cannot be expected (or should not be forced) to be convex or concave, it is necessary to use some general nonlinear optimization techniques and run them from multiple initial solutions to obtain the best possible estimate of the location of the global optima. This method, however, requires the initial solutions to be dense enough in the experimental region, which is feasible only for functions of one or two regressors, becoming unrealistic when the dimension of the experimental region is higher. For this reason, **OptimaRegion** provides separate functions for quadratic polynomials in 2 regressors and for higher oder polynomials in more regressors.

Three factor coverage analysis. For higher dimensional polynomial models, we utilize the GloptiPoly algorithm (Lasserre 2001; Henrion and Lasserre 2003) to search for the global optima. This algorithm reduces a generally non-convex polynomial optimization problem to a sequence of convex linear matrix inequality problems, which generate a sequence of lower bounds monotonically converging to the global optimum of the original problem. For the small-scale response surface problems described in the literature, the global optima can be reached at low computational cost using this method.

The GloptiPoly algorithm is incorporated into the `GloptiPolyRegion` function to compute confidence regions on optima (it is also implemented as a stand alone function for simpler global optimization problems, see the Appendix). To determine the coverage of the CR obtained by `GloptiPolyRegion` we consider the cubic model in 3 variables:

$$f(x) = \mu(x) + \epsilon = 10 - (x_1 - 1)^2 - (x_2 - 2)^2 - (x_3 - 3)^2 + 0.1x_1^3 - 0.1x_2^3 - 0.1x_3^3 + \epsilon$$

defined in the region $R = \{0 \leq x_i \leq 5 \text{ for } i = 1, 2, 3\}$, which, assuming we wish to maximize

coverage	σ	a	b
0.760	5	1.20	1.00
0.047	5	1.50	1.00
0.008	5	0.50	1.00
0.000	5	0.20	1.00
0.000	5	2.00	1.00
0.795	5	1.00	1.20
0.083	5	1.00	1.50
0.001	5	1.00	0.50
0.000	5	1.00	0.20
0.000	5	1.00	2.00
0.985	**5**	**1.00**	**1.00**

Table 3: Estimated coverages of **non-optimal** points $(a \cdot x_1^*, b \cdot x_2^*)$ using 95% bootstrapped CRs as obtained by `OptRegionQuad` for the maximum of a quadratic polynomial regression model. In all cases, $n(reps.) = 5500(500)$, $N_s = 1000$, and $B = 1000$. The last case ($a = b = 1$) corresponds to the coverage of the true optimum point.

has a maximum at $x^* = (1.23, 1.61, 2.24)$. We simulated realizations of this function with $\epsilon \sim N(0, \sigma = 4)$. The CRs are computed from a noisy sample of size $N = nm = 1500$, where $n = 100$ is the number of unique $x-$ locations generated via a HLS design within R and $m = 15$ is the number of replicates at each location. Table 4 shows the coverage analysis under different $n = 100, 150, 200, 500$ values while $m = 15$ is fixed. For each value of n, we computed 200 CRs each based on $B = 1000$ bootstrapped replications. Under the chosen n values, all simulated coverage probabilities are always above the advertised confidence level 95%. As n increases, both the mean and the standard deviation of the volumes of the CRs decrease, providing a more accurate estimate on the location of the optimal point. Table 5 shows the coverages of some non-optimal points for $n = 500$, which indicate that the CRs obtained are unbiased.

n	coverage	$\overline{\text{volume}}$	sd.(volume)	$\frac{\overline{\text{volume}}}{\max \text{volume}}$ (%)	$\frac{\text{sd(volume)}}{\max \text{volume}}$ (%)
100	0.980	0.33	0.25	0.264	0.200
150	0.960	0.09	0.04	0.072	0.032
200	0.970	0.04	0.02	0.032	0.016
500	0.955	0.01	≈ 0	0.008	≈ 0

Table 4: Estimated coverages of the **optimal point** of a 95% bootstrapped CR as obtained by `GloptiPolyRegion` for the maximum of a cubic polynomial regression model. In all cases, $N_s = 200$, $B = 1000$, $m = 15$, and $\sigma = 4$ were used.

Five factor coverage analysis. Consider again the 5-covariate function in example 6 and Figure 7:

$$f(x) = \mu(x) + \epsilon = 10 - (x_1 - 1.5)^2 - (x_2 - 2)^2 - (x_3 - 2.5)^2 - (x_4 - 3)^2 - (x_5 - 3.5)^2$$
$$+0.1x_1^3 - 0.1x_2^3 - 0.1x_3^3 - 0.1x_4^3 - 0.1x_5^3 + x_2x_4 - x_3x_4 + \epsilon$$

defined in the region $R = \{0 \leq x_i \leq 5 \text{ for } i = 1, \cdots 5\}$, which has a global maximum at $x^* =$ (2.28, 2.44, 1.02, 2.65, 2.54). We simulated realizations of this function with $\epsilon \sim N(0, \sigma^2 = 6^2)$. The CR is computed from a sample of size $N = nm$, where n is the number of unique locations generated via a HLS design within R and $m = 15$ is the number of replicates at each location. Table 6 shows the coverage analysis under different $n = 300, 400, 500$ values while $m = 15$ is fixed. For each value of n, we computed 200 CRs each based on $B = 10000$ bootstrapped replications. As n increases, the estimated coverage converges above the advertised confidence level 95%, and both the mean and the standard deviation of the CR volumes decrease. Table 7 shows the

coverage	a	b	c
0.000	1.2	1.0	0.9
0.000	0.2	1.0	0.8
0.000	1.4	1.0	0.7
0.000	0.0	1.0	1.0
0.000	1.0	1.37	1.0
0.000	1.0	0.68	0.3
0.000	1.0	0.34	0.2
0.000	1.0	1.71	1.0
0.955	**1.00**	**1.00**	**1.00**

Table 5: Estimated coverages of **non-optimal** points $(a \cdot x_1^*, b \cdot x_2^*, c \cdot x_3^*)$ using 95% bootstrapped CRs as obtainted by `GloptiPolyRegion` for the maximum of a cubic polynomial regression model. In all cases, $N_s = 200$, $B = 1000$, $n = 500$, $m = 15$, and $\sigma = 4$ were used. The last case ($a = b = c = 1$) corresponds to the coverage of the true optimum point.

n	coverage	$\overline{\text{volume}}$	sd.(volume)	$\frac{\overline{\text{volume}}}{\max \text{volume}}$ (%)	$\frac{\text{sd(volume)}}{\max \text{volume}}$ (%)
300	0.950	3.286e-1	1.679e-1	1.052e-2	5.373e-3
400	0.965	1.337e-1	5.684e-2	4.278e-3	1.819e-3
500	0.970	8.788e-2	4.067e-2	2.812e-3	1.301e-3

Table 6: Estimated coverages of the **optimal point** of a 95% bootstrapped CR as obtained by `GloptiPolyRegion` for the maximum of a cubic polynomial regression model in 5 regressors. In all cases, $N_s = 200$, $B = 10000$, $m = 15$, and $\sigma = 6$ were used.

coverages of some non-optimal points for $n = 500$, which indicates how the CRs obtained by `GloptiPolyRegion` are unbiased.

3) Coverage analysis of the CR on optima from Thin Plate Spline models.

To examine the coverage provided in the case of Thin-Plate Spline model, the simulated function was:

$$f(x_1, x_2) = \big((x_1-2)^2 + (x_2-2)^2 - (x_1-2) + 2(x_1-2)(x_2-2)\big) \exp(-(x_1-2)^2 - (x_2-2)^2)$$

defined in the region $R = \{0 \le x_1 \le 5, 0 \le x_2 \le 5\}$, which has a global maximum in this

coverage	a	b	c	d	e
0.000	1.2	1.0	0.9	1.0	0.7
0.000	0.2	1.0	0.8	0.9	0.8
0.000	1.4	1.0	0.7	0.8	0.9
0.000	0.0	1.0	1.0	0.7	1.0
0.000	1.0	1.37	1.0	0.7	1.0
0.000	1.0	0.68	0.3	0.8	0.9
0.000	1.0	0.34	0.2	0.9	0.8
0.000	1.0	1.71	1.0	1.0	0.7
0.970	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

Table 7: Estimated coverages of **non-optimal** points $(a \cdot x_1^*, b \cdot x_2^*, c \cdot x_3^*, d \cdot x_4^*, e \cdot x_5^*)$ using 95% bootstrapped CRs as obtainted by `GloptiPolyRegion` for the maximum of a cubic polynomial regression model in 5 factors. In all cases, $N_s = 200$, $B = 10000$, $n = 500$, $m = 15$, and $\sigma = 6$ were used. The last case ($a = b = c = d = e = 1$) corresponds to the coverage of the true optimum point.

n	coverage	$\overline{\text{area}}$	sd.(area)	$\frac{\overline{\text{area}}}{\max \text{area}}$ (%)	$\frac{\text{sd(area)}}{\max \text{area}}$ (%)
100	0.988	12.92	6.00	51.68	24.01
150	0.998	11.76	4.75	47.04	19.00
200	1.000	7.14	2.89	28.56	11.56
250	1.000	6.35	2.42	25.40	9.68
300	0.992	5.25	2.26	21.02	9.05
500	0.998	3.90	1.59	15.61	6.36

Table 8: Estimated coverages of the **optimal point** of 95% bootstrapped CRs as obtained by `OptRegionTPS` for the global maximum of a Thin Plate Spline model. In all cases, $N_s = 500$, $B = 200$, $\lambda = 0.04$, and $\sigma = 0.5$ were used. Maximum area in the search region is $25 = (0, 5) \times (0, 5)$.

region[2] at $(x_1^*, x_2^*) = (1.2542, 1.4634)$. In each Monte Carlo simulation we generate n uniformly distributed random $\mathbf{x}$ values over R with observations $f(\mathbf{x}) + \epsilon$ where $\epsilon \sim N(0, \sigma^2)$ are i.i.d.

Computing a projection bootstrap confidence region for a Thin Plate Spline (TPS) model provides higher than advertised coverages of the optimum point, almost always close to 100 %, but with sizes (areas) that decrease rapidly as more experiments are performed (Table 8). These results were obtained with the `OptRegionTps` function.

As it can be seen in Table 9, the coverage percentage of *non-optimal* points is less than the confidence level $1 - \alpha$, with coverage that decays as we consider non-optimum points farther than the optimum (x_1^*, x_2^*). This indicates the projection bootstrapped confidence regions obtained by `OptRegionTps` are also unbiased.

Conclusions

The **OptimaRegion** R package implements useful methods for the computation and display of confidence regions on the location of the global optima of a fitted response surface subject to linear bounds, either based on a polynomial or a Thin Plate Spline response surface model. The functions for 2-covariates include the option of specifying linear constraints that can define a mixture-amount experiment. The methods are particularly valuable for experimenters who need to fit and optimize response surface models and guarantee that only global optima are considered in the confidence regions, a problem that has plagued this field in the past. Given the inherent difficulties of the underlying global optimization problems, problem size limitations are up to a cubic polynomial in up to 5 variables/regressors, and up to two regressors for a Thin Plate Spline. The methods are distribution free as they are based on Tukey's data depth and bootstrapping. Coverage analysis demonstrates that the resulting confidence regions are valid and unbiased, and while the coverage is conservative, the regions are of rapidly diminishing size in the number of observations. A stand alone implementation in R of the GloptiPoly algorithm for global optimization of polynomial functions subject to bounds is also provided in the package, a function not previously available in R.

[2]Note it has another local maxima and a deep minimum as well within the region of interest.

coverage	a	b
0.860	1.20	1.00
0.950	0.20	1.00
0.780	1.40	1.00
0.518	0.00	1.00
0.922	1.00	1.37
0.914	1.00	0.68
0.690	1.00	0.34
0.554	1.00	1.71
0.992	**1.00**	**1.00**

Table 9: Estimated coverages of **non-optimal** points $(a \cdot x_1^*, b \cdot x_2^*)$ using 95% bootstrapped CRs as obtained by `OptRegionTps` for the maximum of a Thin Plate Spline model. In all cases, $n = 300$, $N_s = 500$, $B = 200$, $\lambda = 0.04$, and $\sigma = 0.5$. The last case ($a = b = 1$) corresponds to the coverage of the true optimum point.

Appendix. The GloptiPoly algorithm and the `GloptiPolyR` function for the global optimization of polynomial equations subject to bounds

The package **OptimaRegion** includes an R implementation of the GloptiPoly algorithm for the global optimization of polynomial models as a stand-alone function. Under certain conditions discussed next (which hold for the cases allowed by **OptimaRegion**) the GloptiPoly method guarantees convergence to a global optimum of a polynomial equation subject to polynomial constraints.

Let $g_0(x) : \mathbb{R}^n \to \mathbb{R}$ be a polynomial of degree d_0. The function "gloptipoly" solves the following optimization problem:

$$(P) \quad \min_{x \in K} \; g_0(x),$$

where $K = \{x \in \mathbb{R}^n : g_i(x) \geq 0, i = 1, ..., m\}$, and d_i is the degree of g_i, $i = 1, ..., m$.

Lasserre (2001) provided a method for solving (P). First, it can be shown that (P) is equivalent to

$$\begin{aligned} (Q) \quad \min_{\mu} \quad & \int g_0(x)\, \mathrm{d}\mu(x) \\ \text{s.t.} \quad & \mu(K) = 1, \end{aligned}$$

where the minimum is taken over all probability measures μ on K. Then, instead of solving (Q) directly, we can solve a semidefinite relaxation of (Q) of order N, denoted by (Q^N), and increase the order of relaxation until we obtain the solution to (Q) and hence the solution to (P). Under a particular assumption, the optimal solutions to (Q^N) are guaranteed to converge monotonically to the optimal solution to (P). The assumption happens to be satisfied when the feasible region K is a convex polytope, which is always the case for the problems considered in **OptimaRegion**.[3]

Before we state (Q^N), we must first give some preliminaries. If g_0 is the coefficient vector associated with $g_0(x)$, then we may write

$$g_0(x) = \sum_{\alpha} (g_0)_\alpha x^\alpha,$$

where α is an index such that $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$ and $\sum_{i=1}^n \alpha_i \leq d_0$. Substituting () into the objective function in (Q) yields a sum of moments. We denote these moments as the variables y_α; that is, $y_\alpha = \int (g_0)_\alpha x^\alpha \, \mathrm{d}\mu(x)$. Instead of using a probability measure μ as the decision variable, the relaxed problem uses the moment vector $y = \{y_\alpha\}$. The theory of moments states that a

[3] See Lasserre (2001) for the statement of the assumption.

necessary, but not sufficient, condition for the components of a vector y to correspond to the values of the moments of some probability measure μ is for certain *moment matrices* (made up of the components in y) to be positive semidefinite. Because these conditions are necessary but not sufficient, the relaxed problem (Q^N) will admit more feasible solutions than (Q) itself.

We now describe these moment matrices. Let

$$\{1, x_1, x_2, ..., x_n, x_1^2, x_1x_2, x_1x_3, ..., x_1x_n, x_2^2, x_2x_3, ..., x_n^2, ..., x_n^N\}$$

be a basis for the space of polynomials up to degree N, which has dimension $s(N)$. The moment matrix $M_N(y)$ of order N is a square matrix of dimension $s(N)$ and is formed as follows. Label the rows and columns of $M_N(y)$ with the terms in (). The $(i,j)^{\text{th}}$ entry of $M_N(y)$ is the moment y_α corresponding to the product of the i^{th} and j^{th} terms in (). For instance, if $n = N = 2$, then

$$M_2(y) = \begin{array}{c} \\ 1 \\ x_1 \\ x_2 \\ x_1^2 \\ x_1x_2 \\ x_2^2 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \end{array} \\ \begin{pmatrix} 1 & y_{1,0} & y_{0,1} & y_{2,0} & y_{1,1} & y_{0,2} \\ y_{1,0} & y_{2,0} & y_{1,1} & y_{3,0} & y_{2,1} & y_{1,2} \\ y_{0,1} & y_{1,1} & y_{0,2} & y_{2,1} & y_{1,2} & y_{0,3} \\ y_{2,0} & y_{3,0} & y_{2,1} & y_{4,0} & y_{3,1} & y_{2,2} \\ y_{1,1} & y_{2,1} & y_{1,2} & y_{3,1} & y_{2,2} & y_{1,3} \\ y_{0,2} & y_{1,2} & y_{0,3} & y_{2,2} & y_{1,3} & y_{0,4} \end{pmatrix} \end{array}.$$

The requirement that $M_N(y)$ be positive semidefinite is always a constraint in the relaxed problem (i.e., even when $K = \mathbb{R}^n$). Each additional constraint (i.e., each g_i, $i = 1, ..., m$) warrants that an additional moment matrix $M_{N-v_i}(g_iy)$ be positive semidefinite, where $v_i = \lceil \frac{d_i}{2} \rceil$, $i = 1, ..., m$. As an example to see how these matrices are formed, again take $N = n = 2$ and $g_1(x) = 2 - x_1 \geq 0$. Since $d_1 = 1 = v_1$, the order of the corresponding moment matrix is $N - v_1 = 1$, and we have

$$M_1(y) = \begin{bmatrix} 1 & y_{1,0} & y_{0,1} \\ y_{1,0} & y_{2,0} & y_{1,1} \\ y_{0,1} & y_{1,1} & y_{0,2} \end{bmatrix}.$$

To get $M_1(g_1y)$, we multiply the corresponding basis terms in $M_1(y)$ (e.g., $y_{1,1}$ corresponds to x_1x_2) by $g_1(x) = 2 - x_1$ and convert back to the corresponding moments y_α:[4]

$$M_1(y) = \begin{bmatrix} 2 - y_{1,0} & 2y_{1,0} - y_{2,0} & 2y_{0,1} - y_{1,1} \\ 2y_{1,0} - y_{2,0} & 2y_{2,0} - y_{3,0} & 2y_{1,1} - y_{2,1} \\ 2y_{0,1} - y_{1,1} & 2y_{1,1} - y2, 1 & 2y_{0,2} - y_{1,2} \end{bmatrix}.$$

The complete relaxed problem of order N is thus

$$\begin{aligned} (Q^N) \qquad \min_y \quad & \sum_\alpha (g_0)_\alpha y_\alpha \\ \text{s.t.} \quad & M_N(y) \succeq 0 \\ & M_{N-v_i}(g_iy) \succeq 0, \quad i = 1, ..., m. \end{aligned}$$

The R package **Rdsdp** is used to solve (Q^N). This package uses a dual-scaling algorithm to solve semidefinite optimization problems whose dual takes the form (Zhu and Ye (2016)):

$$\begin{aligned} \sup \quad & b^T y \\ \text{s.t.} \quad & C - \mathcal{A}y \succeq 0. \end{aligned}$$

[4]The reader is referred to Henrion and Lasserre (2003) for additional details.

The "gloptipoly" function reshapes (Q^N) to have the form of () and then calls **Rdsdp** to solve (). The solution is in the form of a vector of moments $y^* = \{y^*_\alpha\}$ which has length $s(2N)$. Recall that our goal is to solve (P) and thus we need the solution in the form of the vector x. Ordinarily then, we would need a procedure for extracting x from y^*. However, because the domain considered in **OptimaRegion** always consists of linear constraints, the n components of y^* corresponding to the terms $x_1, ..., x_n$ (i.e., $y_{1,0,...,0}, y_{0,1,...,0}, ..., y_{0,...,0,1}$) will together always form a feasible solution to (P) (Henrion and Lasserre (2003)). Let $\overline{y^*}$ be the vector y^* truncated to only these n components, and let q^* be the optimal objective value of (Q^N). If $\overline{y^*}$ is feasible (which it always will be in our case) and if $g_0(\overline{y^*}) - q^* < \epsilon$, where ϵ is some tolerance, then the function "gloptipoly" reports that the optimum has been reached. Otherwise, "gloptipoly" increases N by one and repeats the procedure until the tolerance level is satisfied.

The gloptipoly algorithm has been implemented in function `GloptiPolyR`. To illustrate its use, consider the following quadratic function in 3 variables:

$$f(x) = -1.5x_1 + 2.13x_2 - 1.81x_3 + 7.13x_1x_2 + 3.27x_1x_3 + 2.73x_2x_3 + 4.69x_1^2 + 6.27x_2^2 + 5.21x_3^2.$$

The minimum of $f(x)$ over the region $R = \{-2 \leq x_i \leq 2, i = 1, 2, 3\}$ is $x^* = (0.46, -0.46, 0.15)$. The optimization problem can be formally written as:

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to:} \quad & g_1(x) = x_1 + 2 \geq 0 \\ & g_2(x) = x_1 - 2 \leq 0 \\ & g_3(x) = x_2 + 2 \geq 0 \\ & g_4(x) = x_2 - 2 \leq 0 \\ & g_5(x) = x_3 + 2 \geq 0 \\ & g_6(x) = x_3 - 2 \leq 0 \end{aligned}$$

For this problem, the input for `GloptiPolyR` is the specification of problem (P), which needs to be a list of 7 sub-lists, corresponding to $f(x), g_1(x), g_2(x), ..., g_6(x)$, respectively:

```
P <- list()
p_f <- list()
p_g_1 <- list(); p_g_2 <- list(); p_g_3 <- list()
p_g_4 <- list(); p_g_5 <- list(); p_g_6 <- list()
```

Each of these 7 sub-lists has two elements: (1) a multi-dimensional array, denoted by 'c', and (2) an attribute, denoted by 't'. The multi-dimensional array is generated from the monomial coefficients of the corresponding polynomial function. The rule is to put the coefficient of the $x_1^i x_2^j x_3^k$ term in the $[i+1, j+1, k+1]$ position of the array, and place zeroes in other positions:

```
p_f$c <- array(0, dim = c(3, 3, 3))
p_f$c[2, 1, 1] <- -1.5; p_f$c[1, 2, 1] <- 2.13; p_f$c[1, 1, 2] <- -1.81
p_f$c[2, 2, 1] <- 7.13; p_f$c[2, 1, 2] <- 3.27; p_f$c[1, 2, 2] <- 2.73
p_f$c[3, 1, 1] <- 4.69; p_f$c[1, 3, 1] <- 6.27; p_f$c[1, 1, 3] <- 5.21

p_g_1$c <- array(0, dim = c(3, 3, 3))
p_g_1$c[1, 1, 1] <- 2; p_g_1$c[2, 1, 1] <- 1

p_g_2$c <- array(0, dim = c(3, 3, 3))
```

```
p_g_2$c[1, 1, 1] <- -2; p_g_2$c[2, 1, 1] <- 1

p_g_3$c <- array(0, dim = c(3, 3, 3))
p_g_3$c[1, 1, 1] <- 2; p_g_3$c[1, 2, 1] <- 1

p_g_4$c <- array(0, dim = c(3, 3, 3))
p_g_4$c[1, 1, 1] <- -2; p_g_4$c[1, 2, 1] <- 1

p_g_5$c <- array(0, dim = c(3, 3, 3))
p_g_5$c[1, 1, 1] <- 2; p_g_5$c[1, 1, 2] <- 1

p_g_6$c <- array(0, dim = c(3, 3, 3))
p_g_6$c[1, 1, 1] <- -2; p_g_6$c[1, 1, 2] <- 1
```

Next set the attribute for the objective function as either "min" or "max":

```
p_f$t <- "min"
```

Then set the attributes for the constraint functions as either ">=" or "<=":

```
p_g_1$t <- ">="; p_g_2$t <- "<="
p_g_3$t <- ">="; p_g_4$t <- "<="
p_g_5$t <- ">="; p_g_6$t <- "<="
```

Finally, we construct problem (P) from the 7 sub-lists and use it to call `GloptiPolyR`:

```
P <- list(p_f, p_g_1, p_g_2, p_g_3, p_g_4, p_g_5, p_g_6)
result <- GloptiPolyR(P)
```

`GloptiPolyR` returns the global optimal solution and corresponding objective value:

```
> result
$solution
[1]  0.4603 -0.4645  0.1509

$objective
[1] -0.9765
```

References

Box G, Draper N (1987). *Empirical Model Building and Response Surfaces.* John Wiley & Sons, New York,NY.

Box GEP, Hunter JS (1954). "A confidence region for the solution of a set of simultaneous equations with an application to experimental design." *Biometrika*, **41**, 190– 199.

Brooks R, Hunt J, Blows MW, Smith MJ, Bussiere LF, Jennions MD (2005). "Experimental evidence for multivariate stabilizing sexual selection." *Evolution*, **59**(4), 871–880.

Cahya S, del Castillo E, Peterson JJ (2004). "Computation of Confidence Regions for Optimal Factor Levels in Constrained Response Surface Problems." *Journal of Computational and Graphical Statistics*, **13**(2), 499–518.

Carter WH, Wampler GL, Stablein DM, Campbell ED (1982). "Drug activity and therapeutic synergy in cancer treatment"." *Cancer Research*, **42**, 2963–2971,.

Casella G, Berger R (2002). *Statistical Inference*. Duxbury Press.

Del Castillo E (2007). *Process optimization: a statistical approach*, volume 105. Springer.

Del Castillo E, Cahya S (2001). "A Tool for Computing Confidence Regions on the Stationary Point of A Response Surface." *The American Statistician*, **55**(4), 358–365.

Douglas Nychka, Reinhard Furrer, John Paige, Stephan Sain (2017). "fields: Tools for spatial data." doi:10.5065/D6W957CT. R package version 9.8-3, URL https://github.com/NCAR/Fields.

Henrion D, Lasserre JB (2003). "GloptiPoly: Global optimization over polynomials with Matlab and SeDuMi." *ACM Transactions on Mathematical Software (TOMS)*, **29**(2), 165–194.

Lasserre JB (2001). "Global optimization with polynomials and the problem of moments." *SIAM Journal on optimization*, **11**(3), 796–817.

Myers R, Montgomery D (1995). *Response Surface Methodology: Process and Product Optimization using Designed Experiments*. John Wiley & Sons, New York, NY.

Nychka DW (2000). "Spatial-process estimates as smoothers." *Smoothing and regression: approaches, computation, and application*, **329**, 393.

Peterson J (2008). "A Bayesian approach to the ICH Q8 definition of design space." *J. of Biopharmaceutical Statistics*, **18**, 959–975.

Peterson JJ, Cahya S, del Castillo E (2002). "A General Approach to Confidence Regions for Optimal Factor Levels of Response Surfaces." *Biometrics*, **58**(2), 422–431.

Peterson JJ, Novick SJ (2007). "Nonlinear blending: a useful general concept for the assessment of combination drug synergy." *Journal of Receptors and Signal Transduction*, **27**(2-3), 125–146.

Rao CR (1973). *Linear statistical inference and its applications*. Wiley New York.

Rapkin J, Jensen K, Archer CR, House CM, Sakaluk SK, Del Castillo E, Hunt J (2018). "The geometry of nutrient space–based life-history trade-offs: sex-specific effects of macronutrient intake on the trade-off between encapsulation ability and reproductive effort in decorated crickets." *The American Naturalist*, **191**(4), 452–474.

Tukey J (1975). "Mathematics and the picturing of data." *Proceedings of the internaitonal congress of mathematicians*, **2**, 523–531.

Wahba G (1990). *Spline models for observational data*, volume 59. SIAM.

Wan F, Liu W, Bretz F, Han Y (2016). "Confidence sets for optimal factor levels of a response surface." *Biometrics*, **72**(4), 1285–1293.

Wei B, Lee S (2012). "Second-order accuracy of depth-based bootstrap confidence regions." *Journal of Multivariate Analysis*, **105**, 112–123.

Woutersen T, Ham J (2013). "Confidence sets for Continuous and Discontinuous functions of parameters." *Technical report, University of Arizona*.

Zhu Z, Ye Y (2016). "R Interface to DSDP Semidefinite Programming Library." URL https://www.mcs.anl.gov/hs/software/DSDP.

Affiliation:

Enrique del Castillo, Peng Chen and Adam Meyers
Dept. of Industrial & Manufacturing Engineering and Dept. of Statistics
The Pennsylvania State University
310 Leonhard Bldg.
University Park, PA 16802
USA
Emails: exd13@psu.edu, pfc5098@psu.edu, akm5733@psu.edu

John Hunt
School of Science and Health
Western Sidney University
Locked Bag 1797
Penrith, NSW, 2751
Australia
Email: J.Hunt@westernsydney.edu.au

James Rapkin
Department of Biosciences
Daphne du Maurier Building
University of Exeter
Penryn Campus
Penryn
Cornwall
TR10 9FE
UK Email: jr297@exeter.ac.uk