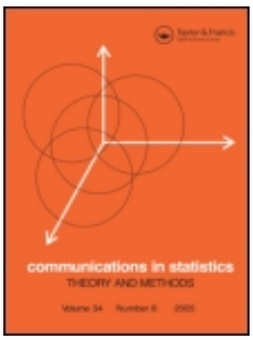

Taylor & Francis
Taylor & Francis Group

Communications in Statistics - Theory and Methods

ISSN: 0361-0926 (Print) 1532-415X (Online) Journal homepage: https://www.tandfonline.com/loi/lsta20

Advances in the Sequential Design of Computer Experiments Based on Active Learning

J. Andrés Christen & Bruno Sansó

To cite this article: J. Andrés Christen & Bruno Sansó (2011) Advances in the Sequential Design of Computer Experiments Based on Active Learning, Communications in Statistics - Theory and Methods, 40:24, 4467-4483, DOI: 10.1080/03610920903518848

To link to this article: https://doi.org/10.1080/03610920903518848

Communications in Statistics—Theory and Methods, 40: 4467–4483, 2011

ISSN: 0361-0926 print/1532-415X online
DOI: 10.1080/03610920903518848

Taylor & Francis
Taylor & Francis Group

Advances in the Sequential Design of Computer Experiments Based on Active Learning

J. ANDRÉS CHRISTEN[1] AND BRUNO SANSÓ[2]

[1]Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Mexico
[2]Department of Applied Mathematics and Statistics, University of California at Santa Cruz (AMS-UCSC), Santa Cruz, California, USA

We present some advances in the design of computer experiments. A Gaussian Process (GP) model is fitted to the computer experiment data as a surrogate model. We investigate using the Active Learning (AL) strategy of finding design points that maximize reduction on predictive variance. Using a series of approximations based on standard results from linear algebra (Weyl's inequalities), we establish a score that approximates the AL utility. Our method is illustrated with a simulated example as well as with an intermediate climate computer model.

Keywords Active learning; Computer experiments; Gaussian processes; Sequential design; Surrogate models.

1. Introduction

Computer experiments have been in use since the dawn of digital computers and may be traced back to the Manhatan project in the 1940's (Feynman, 1985, Ch. 6). They are an increasingly popular method to study complex systems for which direct experimentation is either too costly, too time consuming or simply impossible. Over 750 hits are obtained by a popular scientific search engine with the topic "computer experiment*" (for years 2001–2009 only). Computer experiments involve mathematical models that mimic reality with varying levels of accuracy and complexity. Some require very substantial computing resources involving the use of large supercomputers. In many cases, though, recent advances in computer technology has made it possible for a wide group of researchers to tackle increasingly complex problems with low cost computers.

Computer models usually depend on a number of parameters that may or may not have physical relevance and need to be tuned or calibrated. Performing the computer experiment entails choosing the right combinations of parameter values in an experimental design setting. This task becomes critical when large computing resources need to be allocated for each run. An overview of the field of design and analysis of computer experiments is presented in Stinstra et al.

Received August 27, 2009; Accepted November 27, 2009
Address correspondence to J. Andrés Christen, Centro de Investigación en Matemáticas (CIMAT), Guanajuato, Mexico; E-mail: jac@cimat.mx

(2003). Bursztyn and Steinberg (2006) presented a comprehensive discussion of design criteria in the context of computer experiments. They compare space filling designs, namely latin hypercube, lattice, and rotation designs using various variance and entropy reduction criteria. The authors tend to favor the Integrated Mean Squared Error (IMSE) criterion. This is used to analyze and present algorithmically simple space filling designs (e.g., latin hypercube). They also propose a related but computationally simpler alias matrix criterion. Latin hypercube designs are also favored by other authors. For example, Lehman et al. (2004) elaborated on a design presented by Williams et al. (2000) who used a latin hypercube design as initial trial. In a second stage, the posterior expected improvement is used to generate a design that maximizes a desired criteria (e.g., maximize some aspect of the computer output). Stinstra et al. (2003) also presented algorithms for obtaining maximin designs in a computer experiment context (see also Mease and Bingham, 2006; Fang and Li, 2002; Cioppa and Lucas, 2007, and references therein).

When designing a computer experiment one wishes to "spread" the inputs so as to learn about the model for a wide diversity of parameter configurations (fill the space). A somewhat competing goal is that of focusing on the areas of the parameter space that correspond to high output variability. This approach corresponds to criteria-based designs (see Stinstra et al., 2003, Ch. 6, for a review). Once an information criterion is selected, the design proceeds in a sequential setting in which points are chosen by learning from the output obtained by previously selected parameter values. The criterion proposed in Cohn (1996) consists of maximizing the average reduction in (predictive) variance at every point of a grid when adding a new design point. This is referred to as Active Learning and arises in robotics (we call it Active Learning Cohn or ALC, following Gramacy, 2005). In a Bayesian setting, IMSE is equal to the integrated predictive variance (see Bursztyn and Steinberg, 2006) which in turn may be approximated by the average predictive variance over a grid, which would be equivalent to maximizing the ALC criterion. Therefore, IMSE, or equivalently ALC, seem to be favored by various independent sources.

Applying the ALC sequentially requires the calculation of the change in predictive variance every time a new point is added to the design. Such calculations may be extremely computationally demanding when a fine grid of parameter values is used. If a Gaussian process (GP) is used to approximate the computer output, an increasingly large covariance matrix needs to be inverted for every new point considered. Gramacy and Lee (2008) sifted down a large Latin Hypercube Sample to obtain a small number of candidate locations that are well-spaced relative to themselves, and to the current sampled data points. They then apply the ALC strategy within that small amount of candidate points. The idea is to use the latin hypercube to learn broadly about the parameter space and then use ALC to focus on higher uncertainty regions. In this article, we concentrate on making a new design criterion, based on an approximation to the ALC criterion, that can be easily calculated over large grids. Our proposed design strategy can be easily updated sequentially. It tends to fill the space when small numbers of parameter combinations are available but will favor higher uncertainty regions as additional data become available. Additionally, it may be easily and effectively modified to account for specific features of the computer experiment, like obtaining a design for maximum output or, as in our case study, choosing parameter configurations in the presence of substantive prior knowledge.

In Sec. 2, we present the general Gaussian Process setting to be used in the article. In Sec. 3, we present the design problem and explain our score and sequential scheme. To show the performance of our method in Secs. 4 and 5, we present a simulated 2D example and a direct comparison with ALC designs, respectively. In Sec. 6, we present a case study arising from a design for an intermediate climate computer model. Finally, in Sec. 7 we present a discussion of this article.

2. GP as Surrogate Models

Assume we have a computer model that produces output $z(\boldsymbol{x}) \in \mathbb{R}$ for $\boldsymbol{x} \in \mathcal{D} \subset \mathbb{R}^q$. We suppose that evaluating the computer model, and perhaps post processing the output, is very costly. Following a popular approach (see Stinstra et al., 2003, Ch. 2), we fit a GP to $z(\cdot)$ to minimize the number of evaluations of the actual computer model $z(\cdot)$.

Letting $\boldsymbol{z} = (z(\boldsymbol{x}_1), z(\boldsymbol{x}_2), \ldots, z(\boldsymbol{x}_m))'$ and $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_m)'$

$$\boldsymbol{z} = \boldsymbol{F}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

with $\boldsymbol{\epsilon} \sim N_m(\boldsymbol{0}, \sigma^2 \boldsymbol{R}_{\boldsymbol{\lambda}})$, where $\boldsymbol{R}_{\boldsymbol{\lambda}}$ is the correlation matrix arising from the covariance function $\mathrm{cov}(y(\boldsymbol{x}_i), y(\boldsymbol{x}_j)) = \sigma^2 K_{\boldsymbol{\lambda}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, that depends on a low dimensional parameter $\boldsymbol{\lambda}$. $\boldsymbol{F}$ is a $m \times q$ matrix of regression functions and $\beta \in \mathbb{R}^q$ a set of linear parameters. In a Bayesian setting one defines a prior for the unknown parameters $f(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\lambda})$ to obtain the posterior $f(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\lambda} \mid Data)$. For fixed $\boldsymbol{\lambda}$, a non informative prior for $\boldsymbol{\beta}$ and σ^2 and a standard conjugate calculation leads to a multivariate Normal-Gamma posterior for $f(\boldsymbol{\beta}, \sigma^2 \mid \boldsymbol{\lambda}, Data)$. For unknown $\boldsymbol{\lambda}$, choosing a non informative prior can be problematic. Also, MCMC methods for the exploration of the joint posterior can have very low mixing. Our experience with the t-walk proposed in Christen and Fox (2009) has been very positive.

3. Designing the Experiment

3.1. *Active Learning*

The criterion proposed in Cohn (1996) amounts to a utility function that is given as the average decrease in predictive variance of a GP, for each of the points in a designated grid, if the candidate point is included in the design. Let $\boldsymbol{D}^* = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m)$ be points on a grid; design points are to be taken from this grid. Let $\boldsymbol{D}_N = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$ be points were the computer model has been evaluated. The $(N+1)$-st point is chosen by maximizing

$$ALC(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N) = \frac{1}{m} \sum_{j=1}^{m} V(\boldsymbol{y}_j \mid \boldsymbol{D}_N) - V(\boldsymbol{y}_j \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1}),$$

over the set $\boldsymbol{D}^* p \boldsymbol{D}_N^c$, were $V(\boldsymbol{y} \mid \boldsymbol{D}_N)$ is the predictive variance of the GP at $\boldsymbol{y}$ and $V(\boldsymbol{y} \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1})$ is the expected predictive variance at $\boldsymbol{y}$. As explained in Sec. 1, in a Bayesian setting, ALC is approximately equivalent (over the grid points) to minimizing the Integrated (predictive) Mean Square Error. This in turn is equivalent, under certain circumstances, to other design criteria (see Bursztyn and Steinberg, 2006, and references therein). After a new design point is found, ideally

the corresponding computer model output is calculated and new predictive variances are obtained to search for the next design point. ALC is thus intrinsically sequential. It is reported to produce good designs but, unfortunately, for a GP, evaluating $ALC(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N)$ for every point in a large grid is simply not feasible. Thus strategies to reduce the search, like the one implemented in Gramacy and Lee (2008), need to be considered.

3.2. *An Alternative to ALC*

We present a new design score that is based on a cheap approximation to ALC, based on a lower bound for the reduction in predictive variance (see the Appendix for a formal argument). Let $c(\boldsymbol{y}, \boldsymbol{x}) = \text{cov}(\boldsymbol{y}, \boldsymbol{x} \mid \boldsymbol{D}_N)$ and $V(\boldsymbol{y}) = \text{cov}(\boldsymbol{y}, \boldsymbol{y} \mid \boldsymbol{D}_N)$ for ease of notation. The score is defined as

$$A(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N) = \frac{1 - \|r(\boldsymbol{x}_{N+1})\|}{m\sqrt{V(\boldsymbol{x}_{N+1})}} \sum_{j=1}^{m} \frac{c(\boldsymbol{y}_j, \boldsymbol{x}_{N+1})^2}{\sqrt{V(\boldsymbol{y}_j)}}, \tag{1}$$

where $\|r(\boldsymbol{x}_{N+1})\|^2 = \sum_{i=1}^{N} \frac{c(\boldsymbol{x}_{N+1}, \boldsymbol{x}_i)^2}{V(\boldsymbol{x}_{N+1})V(\boldsymbol{x}_i)}$. For a heuristic justification, we note that

$$A(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N) = \frac{1 - \|r(\boldsymbol{x}_{N+1})\|}{m} \left(V(\boldsymbol{x}_{N+1}) + \sum_{\boldsymbol{y} \neq \boldsymbol{x}_{N+1}} \frac{c(\boldsymbol{y}, \boldsymbol{x}_{N+1})^2}{\sqrt{V(\boldsymbol{y})V(\boldsymbol{x}_{N+1})}} \right). \tag{2}$$

Therefore, the score chooses a point $\boldsymbol{x}_{N+1}$ with high predictive variance that is correlated with other points in $\boldsymbol{D}^*$. This is a desirable feature, as $\boldsymbol{x}_{N+1}$ will provide information about points correlated with it. Additionally, $1 - \|r(\boldsymbol{x}_{N+1})\|$ prevents $\boldsymbol{x}_{N+1}$ to be too close to points in $\boldsymbol{D}_N$. Note that the score has the same units as $V(\cdot)$, as it is the case for ALC. For zero correlation, our score becomes $\frac{1}{m}V(\boldsymbol{x}_{N+1})$ and points are allocated to maximize the predictive variance; under certain circumstances this is also the case for ALC, see the Appendix for details.

For comparison to a space filling strategy, like for example maximin designs, consider the term $1 - \|r(\boldsymbol{x}_{N+1})\|$. If we restrict our attention to correlations such that $K(\boldsymbol{x}_{N+1}, \boldsymbol{x}_i)^2$ decreases as the distance between $\boldsymbol{x}_{N+1}$ and $\boldsymbol{x}_i$, maximizing this term only will be equivalent to minimizing $\frac{1}{N}\|r(\boldsymbol{x}_{N+1})\|^2$. Then, maximizing $1 - \|r(\boldsymbol{x}_{N+1})\|$ will have similar results as maximizing the average distance to already selected points, a similar strategy as maximizing the minimum distance to selected points (a maximin design). Thus, the space filling property of our design. The second factor in Eq. (2) makes our design sensitive to the underlying variance structure in our problem, concentrating sampling where there is higher uncertainty.

Note from (1), that computationally speaking, our score is of a maximum order of $O(m^2)$, where m is the grid size. This assuming that we have an estimator of $\boldsymbol{\lambda}$ for the current sample and that an evaluation of the covariance function is of order $O(1)$ (it is clear from (1) that our score only requires evaluation of $c(\cdot, \cdot)$, m^2 times). Under the same assumption, ALC is of order $O(m^2N^3)$ where N is the current sample size, since inverting the $N \times N$ correlation matrix to obtain $V(\cdot \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1})$ is of order N^3. However, Gramacy and Lee (2009) showed, using the Barnett partition, that only one $O(N^3)$ operation needs to be performed and the subsequent ALC evaluations run in N^2 order; therefore, ALC can be performed in order $O(m^2N^2 + N^3)$. Computationally speaking A is a substantial improvement over ALC.

A popular assumption in computer modeling is that of separability of the correlation function. This implies that the correlation can be factorized as $K_{\boldsymbol{\lambda}}(\boldsymbol{x}, \boldsymbol{x}) = \prod_{j=1}^{q} k^{j}_{\lambda_j}(x_j, y_j)$. In some applications it might be the case that, for some j, $\boldsymbol{\lambda}_j$ is estimated as making $k^{j}_{\boldsymbol{\lambda}_j}$ almost irrelevant for the response, leading to a correlation that is independent of the jth factor. It is easily seen from Eq. (1) that our score will be constant along the jth factor. In this case points will be selected as to maximize the score in the relevant factors and randomly in the null factor.

3.3. *Sequential Procedure*

Once we select a new design point $\boldsymbol{x}_{N+1}$ using the proposed score, we evaluate the computer model at $\boldsymbol{x}_{N+1}$. Given the output $z(\boldsymbol{x}_{N+1})$, we re-estimate the parameters of the surrogate model and reevaluate the covariance to $\mathrm{cov}(\cdot, \cdot \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1})$. To move to the next step, we use again the score to obtain a new design point $\boldsymbol{x}_{N+2}$. However, in some applications, several new design points need to be processed in batches and we cannot wait for the output $z(\boldsymbol{x}_{N+1})$ to find the next design point. In such a case, we could naively add points to the design and recalculate the score. The resulting batches, however, tend to occupy only current high variance regions according to the most recent estimate of the correlation structure, and in our experience the resulting designs are simply not satisfactory.

Updating the correlation structure somehow, even in the case that we cannot wait for the output $z(\boldsymbol{x}_{N+1})$, is imperative to obtain a reasonable sequential procedure. In the GP setting as above, assuming that we have fixed the correlation parameters to an estimator $\hat{\boldsymbol{\lambda}}(\boldsymbol{D}_N)$, the predictive variance $V(\cdot \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1})$ does not depend on the actual observed value $z(\boldsymbol{x}_{N+1})$ but only on the design point $\boldsymbol{x}_{N+1}$; therefore, we do not need the actual response $z(\boldsymbol{x}_{N+1})$ to update the correlation structure. Once responses are obtained for k additional points, the correlation parameters are reestimated to obtain $\hat{\boldsymbol{\lambda}}(\boldsymbol{D}_{N+k})$ and $\mathrm{cov}(\boldsymbol{y}_i, \boldsymbol{y}_j \mid \boldsymbol{D}_{N+k})$. This provides a sequential scheme, which we use in our simple 2D example in Sec. 4 and the computer model example in Sec. 6.

3.4. *Purpose Specific Designs*

Suppose there is a feature $u(\boldsymbol{x})$ of the computer model and/or of the experimental region that one wishes to maximize. In other words, we need to explore the parameter space to find the maximum of u. It is possible that some parts of the design space with low $u(\boldsymbol{x})$ values include rugged parts of the output that would correspond to high predictive variance for the surrogate GP model. High variance will call for more design points in those regions, but evaluating the computer model at points with low $u(\boldsymbol{x})$ would be irrelevant for the maximization of u. Stinstra et al. (2003, Ch. 6) presented a review of some optimization driven design criteria. Most of the proposed designs require an initial learning stage, that typically uses a space filling design. After the initial stage, a purpose specific search is conducted. Various levels of complexity in the search algorithms are described by Stinstra et al. (2003) and Schonlau et al. (1998). Here we present a very simple alternative, that seamlessly progresses from the learning stage to the purpose specific design.

We propose to bias our score in Eq. (1) to focus on the regions that are relevant to the maximization of $u(\boldsymbol{x})$. The proposed biased score is

$$\{u(\boldsymbol{x}_{N+1})\}^{\frac{N}{wm}} A(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N)$$

for some positive w (we require $0 \leq u(\boldsymbol{x}) \leq 1$). For small N, the bias term is negligible and the score will spread design points on the grid according to the criteria discussed in Sec. 3.2. For large N, the bias term will black out points with low $u(\boldsymbol{x})$ values, thus concentrating the design points on maximizing u. For example, if the design's main purpose is maximization of computer model output we could let $u(\boldsymbol{x}) = \hat{z}(\boldsymbol{x})$, the current predictive value at $\boldsymbol{x}$. The score then becomes $\{\hat{z}(\boldsymbol{x}_{N+1})\}^{\frac{N}{wm}} A(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N)$. This will first behave basically as $A(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N)$, but will black out low values of $z(\cdot)$, when N is big (since $0.1 \leq u \leq 1$, $u^x \approx 1$ for x small). Increased variability is observed as x increases to 1, therefore, low values of u will progressively decrease the score but high values of u will basically lead to $A(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N)$. If we envisage a maximum design size $N = \alpha m (0 < \alpha \leq 1)$ the "discount" in $A(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N)$ is $d = u^{\frac{z}{w}}$ or $w = \frac{\alpha \log(u)}{\log(d)}$. For example, for a design with $\alpha = 1/5$ of the grid size, and $u \leq 0.1$, less than 10% of the maximum value, a minimum discount factor $d = 0.9$ will be obtained with $w \approx 5$; only very low values of u will be heavily discounted.

In Sec. 6, we use the idea of biasing the score in the case where the feature u is proportional to a prior density for $\boldsymbol{x}$, $u(\boldsymbol{x}) \propto f(\boldsymbol{x})$. This seems appropriate for calibration problems where there is good prior information about the most likely values in the parameter space. One may also try to use a latin hyper cube design with $f(\boldsymbol{x})$ as underlying measure. This will produce spread out design points with high prior density. But no information about the variability of the field is incorporated.

4. Simulated Example

To test our score, we simulated data according to the model $z(x_1, x_2) = (x_1 - 4.5) + (x_2 - 3.5) + e$, where $e \sim N(0, 3^2)$. We considered a regular grid of 5,000 points in the region $[-2, 6] \times [-2, 6]$ and used a GP model with a linear regressor, as explained in Sec. 2, with correlation function $\exp(-d/\lambda)$, where d is the distance between any two points. λ is fixed to either 0.3, 0.4, or 0.8. We took 5 points regularly spaced on the grid as initial trial and then calculated 30 design points using our proposed score with no bias. Figure 1(a) presents a Latin hypercube design (taken from the internet, see figure caption) that by definition will ignore the correlation structure providing a regular space filling design. Figure 1(b) presents a maximin design (see Sec. 3.2). Figures 1(c), (d), and (e) present our design under the three different values of λ. Figure 1(f) shows the results of taking the first 10 design points in Fig. 1(e), evaluated the surface at those points, recalculate the model parameters and calculate 20 additional design points. So the final design consists of 30 points in both cases. Apart from small changes in the ordering, due to numerical error, the 20 remaining design points are the same in both cases. This is a most desirable sequential behavior. Gray levels in Fig. 1 represent predictive variance.

Some interesting features of our design are to be highlighted. The points are well spread across the region. For low correlation (Fig. 1(c)), points are allocated relatively close to each other and tend to occupy high predictive variance regions.

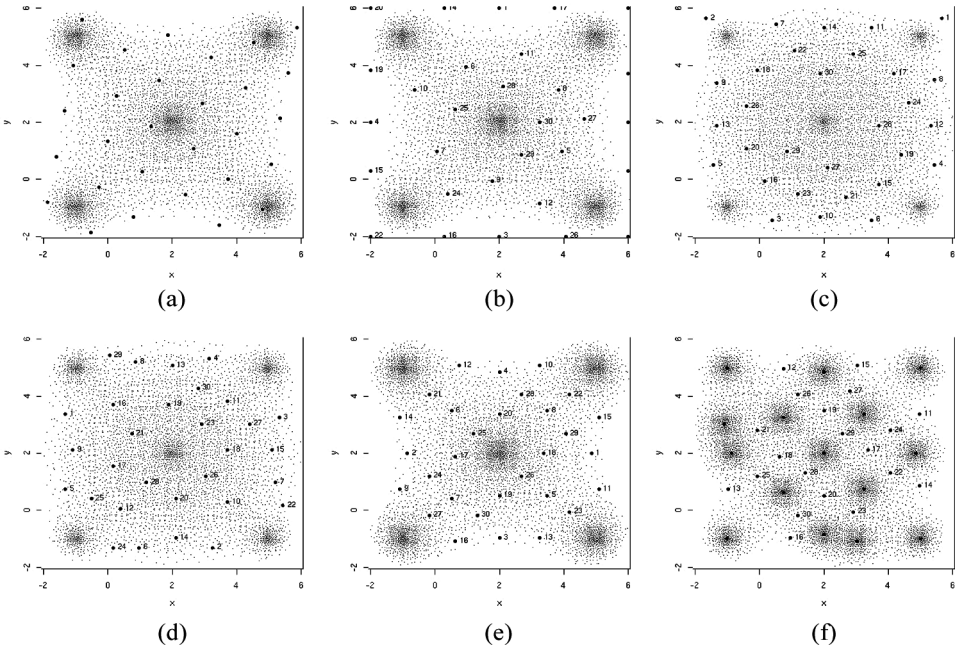

Figure 1. Simulated data from the model $(x_1 - 4.5) + (x_2 - 3.5)$ with 5 regularly spaced points taken initially from a grid of 5,000 points. 30 points were added from: (a) a L2 Latin hypercube design (see http://www.spacefillingdesigns.nl/maximin/mml2lhd2d_designs/mml2lhd2d030.html); (b) a maximin design; (c) our design, with correlation parameter $\lambda = 0.3$; (d) with correlation parameter $\lambda = 0.4$; an (e) with correlation parameter $\lambda = 0.8$; and (f) same as (e) but with updated parameters after adding the first 10 points. In all cases, topo colors represent predictive variance.

The fact that point allocation is sensitive to the correlation, as seen when comparing Figs. 1(c), (d), and (e), is a particularly desirable feature.

Note that the highest predictive variance occurs at the edge of the region, but no points are allocated to the edges. This is due to the tradeoff between high predictive variance and high correlation with other points on the grid, as discussed in Sec. 3.2.

5. Comparisons with ALC

As discussed in Steinberg and Bursztyn (2004), in any GP there is a trade off between the variability that is captured by the linear terms an that captured by the covariance function. Therefore we expect that the presence of regressors will affect the behaviour of our score. As our score is informaly based on a lower bound for the ALC, we expect that $\boldsymbol{F}$ will affect our score and ALC differently. We present here a comparison. In Fig. 2 we compare the actual (relative) values for the scores using a constant, linear $(x - 2) + (y - 2)$ and quadratic $(x - 2)^2 + (y - 2)^2$ functions as true models. In each case, the GP regressor is set to include the true model up to unknown coefficients. The top row in Fig. 2 corresponds to the constant true model using a constant regressor. As expected, the score values of ALC and A have similar shapes and the actual maxima are rather close.

However, as seen in the middle and bottom rows of Fig. 2, the scores differ in their shapes, leading to different maxima. Certainly, the regressor does make a difference in ALC. Also, it makes a difference in our score A, indirectly through the predictive covariance used. However, ALC is strongly influenced by the regressor and A has a more "neutral" behavior.

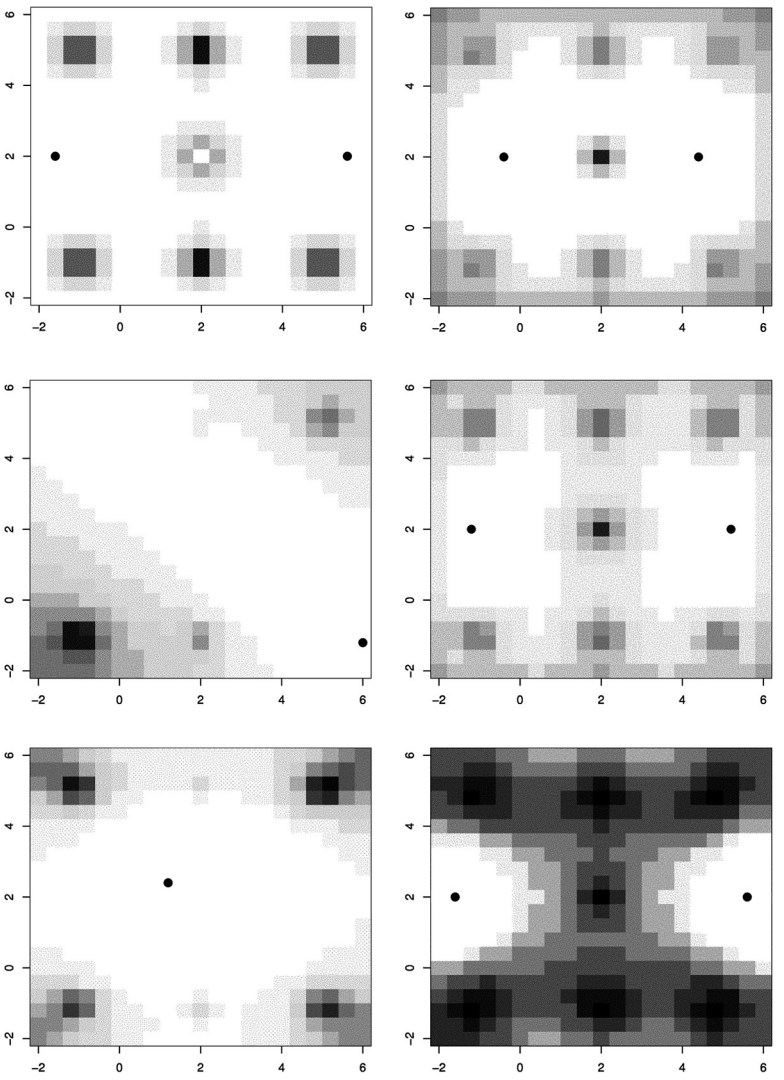

Figure 2. Comparison of the relative score values for ALC (left panels) and our score (right panels), with six initial points. The underlying true model is a constant (top row), linear $((x-2)+(y-2)$, middle row), and quadratic $((x-2)^2+(y-2)^2$, bottom row). In each case, the GP regressor model includes the true model. Blue dots represent the corresponding maxima (chosen design point(s) in each case). We use an exponential decay correlation function with parameter 0.7 as in Sec. 4. Note that topo colors here represent relative (ALC and A) score values.

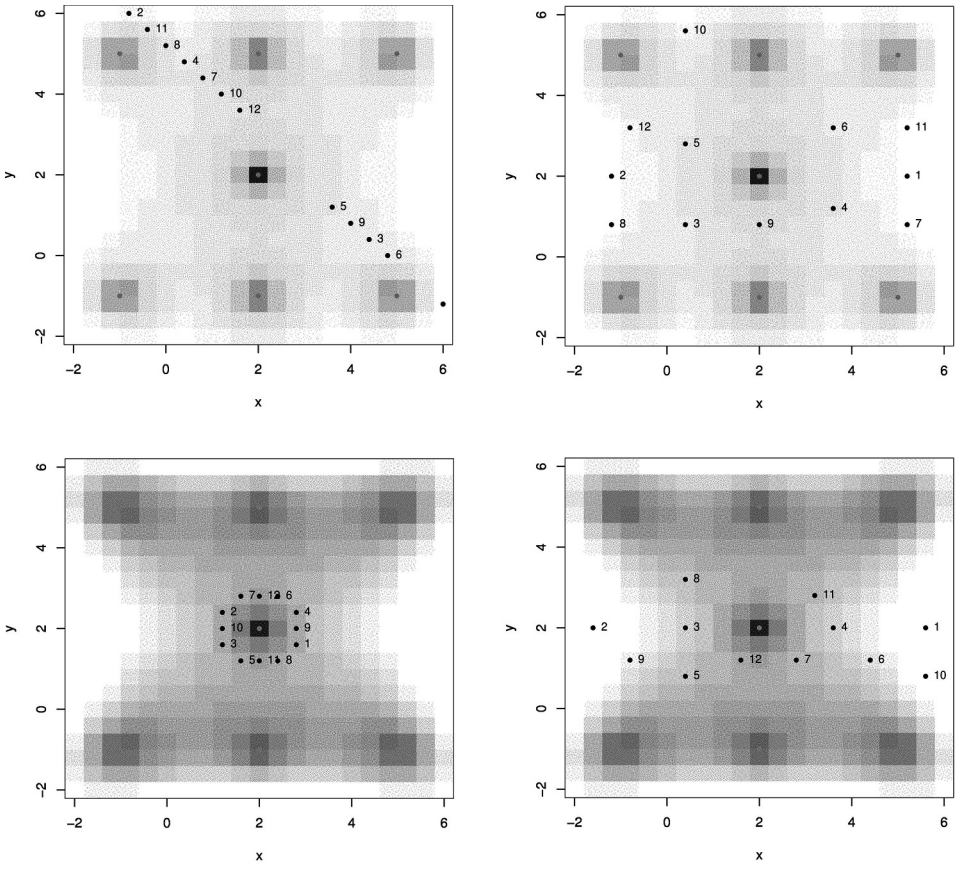

Figure 3. Comparison of 12 design points for ALC (left panels) and our score (right panels), with 6 initial points. The underlying true model is linear (top row) and quadratic (bottom row). In each case, the GP regressor model includes the true model. Dots represent the corresponding design points. We use an exponential decay correlation function with parameter 0.7 as in Sec. 4. Note that topo colors here represent relative (ALC and A) score values.

Another comparison is presented in Fig. 3, using the linear and quadratic models as above, and producing 12 design points both with ALC and with our score A. Again, our score produces different but more neutral designs. ALC is heavily dependent on the regressor. As far as reducing variance is concerned, ALC is better (again, by definition), but perhaps more spread in the design points will permit a better exploration of the experimental space.

6. Case Study

We consider the problem of analyzing the output of an intermediate climate computer model. We focus on the MIT2D Climate Model (MIT2DCM) described in Sokolov and Stone (1998) and Forest et al. (2006). The MIT2DCM provides simulations of the ocean, surface and upper atmospheric temperature over a grid of 46 different latitudes and 11 vertical layers. The model was run for the years

1860–1995 and its output was summarized in three low dimensional statistics. We use the summary consisting of the trend of deep ocean temperatures obtained from the pentadatal averages for the 0–3 km deep layer during the period 1952–1995. The model output is controlled via three parameters, $(x_1, x_2, x_3) = \boldsymbol{x}$. These parameters are:

- x_1 – "The rate of diffusion for heat anomalies into the deep-ocean, $\mathcal{K}_v$".
- x_2 – "Climate sensitivity, $\mathcal{S}$", defined as the equilibrium global mean surface temperature response to a doubling of CO_2.
- x_3 – "The net anthropogenic aerosol forcing $\mathcal{F}_{aer}$".

The MIT2DCM model was evaluated at an irregular grid consisting of 426 combinations of these parameters. Further details on the statistical analysis of the MIT2DCM can be found in Sansó et al. (2008) (there is a very relevant calibration question for this problem but here we will focus solely on analyzing the computer model output and designing the experiment).

We have very good prior information about the most relevant values of $\boldsymbol{x}$ (in the sense of being closer to the real but unknown values for $(\mathcal{K}_v, \mathcal{S}, \mathcal{F}_{aer})$) from the literature about climate change. This is used to build the distribution $\pi(\cdot)$ of relevant values for $\boldsymbol{x}$. For $\sqrt{x_1}$ we consider a beta distribution with support on (0, 6) and parameters (3.5, 6). For x_2, we use a beta distribution with support (0, 15) and parameters (2.85, 14) and for x_3 we use a beta distribution supported on $(-1.5, .5)$ with parameters (4, 4). We assume that, *a priori*, the three parameters are independent. A more detailed discussion of these choices appears in Sansó et al. (2008).

We propose a criterion that biases our score towards the region of high-density values for $\pi(\cdot)$ (as explained in Sec. 3.4). This is achieved by the score

$$\left\{ \frac{\pi(\boldsymbol{x}_{N+1})}{\max \pi(\boldsymbol{x})} \right\}^{\frac{N}{wm}} A(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N).$$

This score depends on the quantity w which needs to be appropriately tuned in relation to the ratio of actual computed points N to possible grid values m. For the MIT2DCM example we took 30 design points at random from the 426 original grid as initial trial. We then calculated our score to obtain 100 design points. This was done both on the original grid and in a larger regular grid. We proceeded sequentially as explained in section 3.3, using the correlation function $K_{\boldsymbol{\lambda}}(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{3} \exp(-\lambda_i |x_i - y_i|)$. $\boldsymbol{\lambda}$ was estimated using MCMC. The value of w was set to 5 after some tuning (see Sec. 3.4).

We consider a regular grid to generate our designs. Each variable is divided into 10 equally spaced points in its prior range, and the grid is the resulting product set obtaining $10^3 = 1,000$ points. For the MIT2D climate model, an irregular grid of 426 points was used. The irregular grid was created, heuristically, biased by the prior information on $\boldsymbol{x}(\pi(\boldsymbol{x}))$. A better observation of our score performance is thus expected if we use our regular grid. However, we needed a trial sample to make an estimation of the covariance structure and model responses are only available in the original grid. Accordingly, we took 30 points at random form the original grid as initial trial (the gray points, outside the regular grid, seen in Figs. 4–7). After analyzing the residuals obtained by a linear model, we decided to use the regressor $x_1 + x_2 + x_3 + x_1^2 + x_2^2 + x_3^2$ obtaining no evident trend in the residuals.

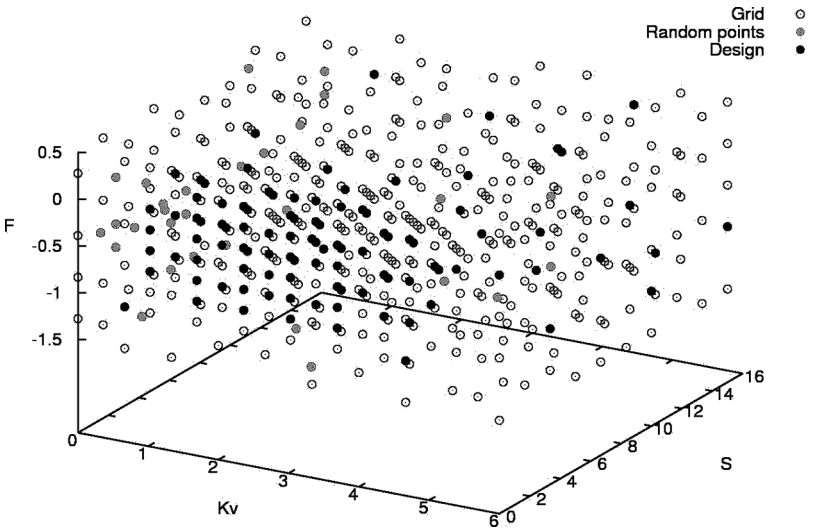

Figure 4. Regular grid (1,000 points) with 30 random points taken as initial trail and 100 design points from our score weighted by the prior.

Figure 4 presents a 3D view of the resulting design weighted by the prior $f(\boldsymbol{x})$, using a regular grid, with 30 initial random trail points and 100 design points. Also interesting are the two-by-two projections seen in Fig. 5. Note that for 130 points, less than $1/3$ of the original sample, nearly the same predictive variance is expected as with the full sample (see the left bottom panel). Also, most of the two by two projections are covered using 30% of the available grid points.

For comparisons we also produced a design using the original score, not weighted by the prior $f(\boldsymbol{x})$. In Fig. 6, we present such design and in Fig. 7 the corresponding two-by-two projections. Comparing Figs. 4 and 6 and Figs. 5 and 7 it is clear how more design points are selected in higher prior density regions in the former than in the latter (as expected), but nevertheless points are still spread across the design region. In both cases, using 130 points most of the predictive variance is covered, perhaps questioning the need for the costly 426 point run originally performed. In all cases, similar results are obtained considering the original grid used in the MIT2D climate model (results not shown).

7. Discussion

We have presented a new score to evaluate points to generate sequential designs, mainly in the context of GP models used as surrogate substitutes for complex computer models. This score is loosely based on the Active Learning strategy used in robotics (Cohn, 1996). We also generalized the sequential procedure when several design points are needed before the computer output is available. In our tests, the score performed quite well, producing reasonable designs with relatively low computational burden. Besides the motivating formal argument (based on standard linear algebra results), our design score is also intuitively clear and may be generalized to other spacial processes. The only requirements are a current estimate of the covariance function and an initial grid of points to base the design.

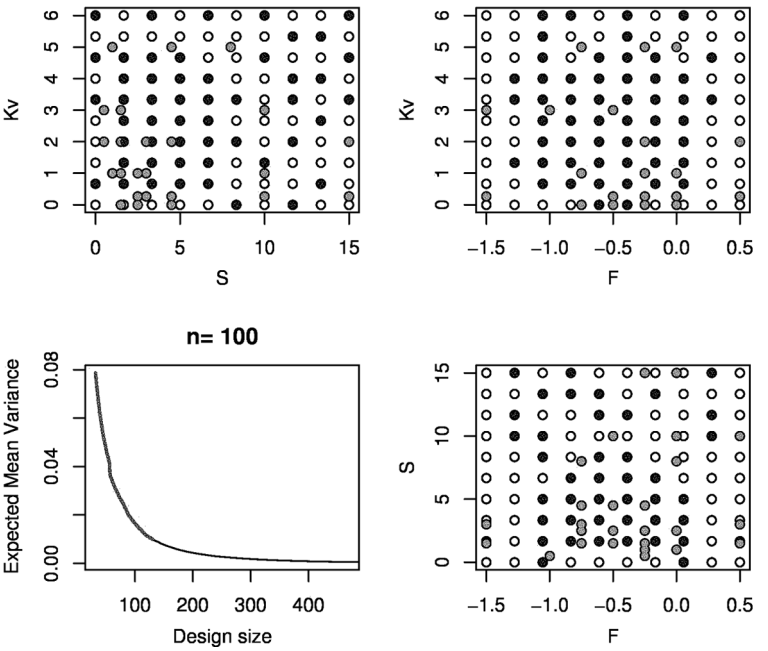

Figure 5. Two by two representation of our design, using a regular grid (1,000 points), with 30 random points taken as initial trail and the 100 design points taken using our score weighted by the prior. Note how with only 130 points most projections are covered. In the left bottom panel we present the expected average predictive variance for all design sizes. With this 130-point design the plot is highlighted; the expected variance is nearly the same as for the full 426-point sample.

Specifically, we would like to experiment using our design scheme for a surrogate model based on non-stationary treed Gaussian Processes (Gramacy and Lee, 2008).

We have experimented with the use of our biased design strategy for maximization purposes. For the example presented in Sec. 4, we obtained very good results. Design points are first scattered around the design region (very much as presented in Fig. 1) but are progressively assigned around the maximum. The global maximum is soon found using very few evaluations of the objective function. These and other potentially favorable characteristics of our design scheme should be further investigated and are left for future research.

A Formal Argument for Equation (1)

In a GP as in Sec. 2, given the inverse of the correlation matrix $\boldsymbol{R}_{\boldsymbol{D}_N}^{-1}$, σ^2, and $\boldsymbol{\beta}$ we have that

$$V(\boldsymbol{y} \mid \boldsymbol{D}_N) = \sigma^2(1 - r(\boldsymbol{y})' \boldsymbol{R}_{\boldsymbol{D}_N}^{-1} r(\boldsymbol{y})),$$

where $r(\boldsymbol{y})' = (K(\boldsymbol{y}, \boldsymbol{x}_1), K(\boldsymbol{y}, \boldsymbol{x}_2), \ldots, K(\boldsymbol{y}, \boldsymbol{x}_N))$ (we assume any correlation parameters $\boldsymbol{\lambda}$ to be fixed and write $K_{\lambda}(\cdot, \cdot) = K(\cdot, \cdot)$).

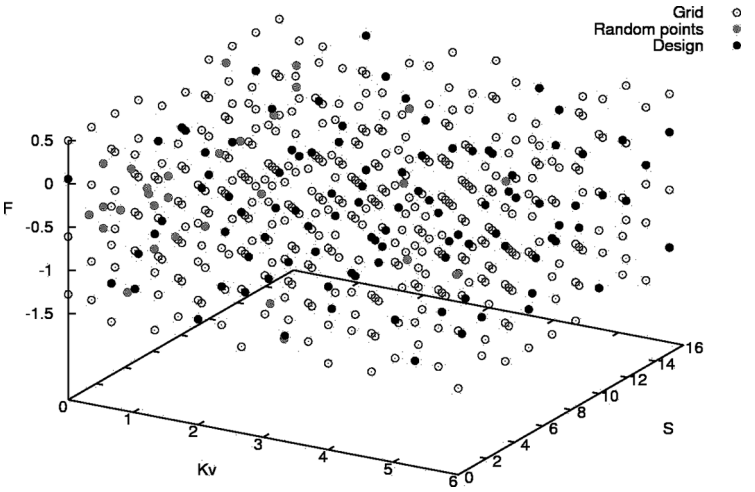

Figure 6. Regular grid (1,000 points) with 30 random points taken as initial trail and 100 design points from our score, not weighted by the prior.

Therefore,

$$ALC(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N) = \frac{\sigma^2}{m} \sum_{j=1}^{m} r_1(\boldsymbol{y}_j)' \boldsymbol{R}_{\boldsymbol{D}_{N+1}}^{-1} r_1(\boldsymbol{y}_j) - r(\boldsymbol{y}_j)' \boldsymbol{R}_{\boldsymbol{D}_N}^{-1} r(\boldsymbol{y}_j),$$

where $\boldsymbol{D}_{N+1} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N, \boldsymbol{x}_{N+1})$ and $r_1(\boldsymbol{y})' = (K(\boldsymbol{y}, \boldsymbol{x}_1), K(\boldsymbol{y}, \boldsymbol{x}_2), \ldots, K(\boldsymbol{y}, \boldsymbol{x}_N), K(\boldsymbol{y}, \boldsymbol{x}_{N+1}))$.

We will establish a lower bound for

$$D(\boldsymbol{y} \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1}) = r_1(\boldsymbol{y})' \boldsymbol{R}_{\boldsymbol{D}_{N+1}}^{-1} r_1(\boldsymbol{y}) - r(\boldsymbol{y})' \boldsymbol{R}_{\boldsymbol{D}_N}^{-1} r(\boldsymbol{y}) \tag{3}$$

in order to obtain a lower bound for $ALC(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N)$.

We write

$$\boldsymbol{R}_{\boldsymbol{D}_{N+1}} = \boldsymbol{R} + \mathbf{E} = \begin{bmatrix} \boldsymbol{R}_{\boldsymbol{D}_N} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} + \begin{bmatrix} \mathbf{0} & r(\boldsymbol{x}_{N+1}) \\ r(\boldsymbol{x}_{N+1})' & 0 \end{bmatrix}.$$

A simple calculation leads to

$$D(\boldsymbol{y} \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1}) = K^2(\boldsymbol{y}, \boldsymbol{x}_{N+1}) + r_1(\boldsymbol{y})' (\boldsymbol{R} + \boldsymbol{E})^{-1} r_1(\boldsymbol{y}) - r_1(\boldsymbol{y})' \boldsymbol{R}^{-1} r_1(\boldsymbol{y}). \tag{4}$$

Let $\lambda_i(\boldsymbol{A})$, $i = 1, \ldots, N+1$ denote the eigenvalues of a matrix $\boldsymbol{A}$ in decreasing order. Let $e_i(\boldsymbol{A})$ denote the corresponding normalized eigenvectors. We assume that $\lambda_{N+1}(\boldsymbol{R} + \boldsymbol{E}) > 0$ and $\lambda_{N+1}(\boldsymbol{R}) > 0$. We see that

$$D(\boldsymbol{y} \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1}) = K^2(\boldsymbol{y}, \boldsymbol{x}_{N+1}) + \sum_{i=1}^{N+1} \lambda_i(\boldsymbol{R} + \boldsymbol{E})^{-1} (e_i(\boldsymbol{R} + \boldsymbol{E})' r_1(\boldsymbol{y}))^2$$
$$- \sum_{i=1}^{N+1} \lambda_i(\boldsymbol{R})^{-1} (e_i(\boldsymbol{R})' r_1(\boldsymbol{y}))^2.$$

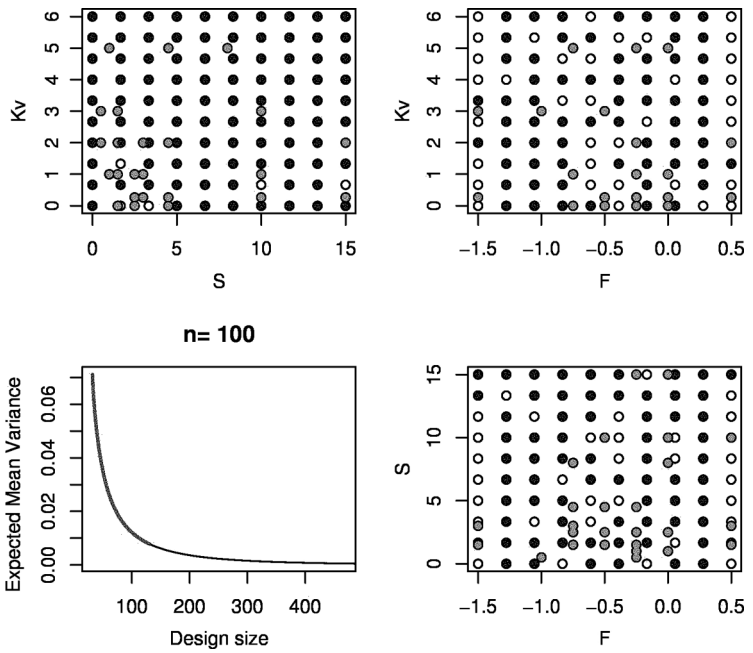

Figure 7. Two by two representation of our design, with a regular grid (1,000 points) with 30 random points taken as initial trail and the 100 design points taken using our score, not weighted by the prior. Note how with only 130 points most projections are covered. In the left bottom panel we present the expected average predictive variance for all design sizes. With this 130-point design the plot is highlighted; the expected variance is nearly the same as for the full 426-point sample.

Assuming that the average change in eigenvectors is small we take

$$\sum_{i=1}^{N+1} \lambda_i(\boldsymbol{R})^{-1}(e_i(\boldsymbol{R})' r_1(\boldsymbol{y}))^2 = \sum_{i=1}^{N+1} \lambda_i(\boldsymbol{R})^{-1}(e_i(\boldsymbol{R}+\boldsymbol{E})' r_1(\boldsymbol{y}))^2, \tag{5}$$

and obtain

$$D(\boldsymbol{y} \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1}) = K^2(\boldsymbol{y}, \boldsymbol{x}_{N+1}) + \sum_{i=1}^{N+1} (\lambda_i(\boldsymbol{R}+\boldsymbol{E})^{-1} - \lambda_i(\boldsymbol{R})^{-1})(e_i(\boldsymbol{R}+\boldsymbol{E})' r_1(\boldsymbol{y}))^2.$$

Indeed, in order to obtain a proper lower bound further justification will be needed for the assumed equality in (5), but the argument as presented is only to show a justification for our score in (1), see below. Note that $(e_i(\boldsymbol{R}+\boldsymbol{E})' r_1(\boldsymbol{y}))^2 = r_1(\boldsymbol{y})' r_1(\boldsymbol{y})(\cos\theta_i)^2$, where θ_i is the angle between vectors $e_i(\boldsymbol{R}+\boldsymbol{E})$ and $r_1(\boldsymbol{y})$. On the other hand,

$$\lambda_i(\boldsymbol{R}+\boldsymbol{E})^{-1} - \lambda_i(\boldsymbol{R})^{-1} = \frac{1-\lambda_i(\boldsymbol{R}+\boldsymbol{E})\lambda_i^{-1}(\boldsymbol{R})}{\lambda_i(\boldsymbol{R}+\boldsymbol{E})} \geq \frac{1-\lambda_i(\boldsymbol{R}+\boldsymbol{E})\lambda_i^{-1}(\boldsymbol{R})}{\lambda_1(\boldsymbol{R})+\lambda_1(\boldsymbol{E})}.$$

The last inequality is part of Weyl's inequalities (Bhatia, 1997, Ch. III) since $\lambda_i(\boldsymbol{R}+\boldsymbol{E}) \leq \lambda_i(\boldsymbol{R}) + \lambda_1(\boldsymbol{E}) \leq \lambda_1(\boldsymbol{R}) + \lambda_1(\boldsymbol{E})$. By direct calculations we see that

$\lambda_1(\boldsymbol{E}) = \|r(\boldsymbol{x}_{N+1})\|$. Thus, we obtain

$$D(\boldsymbol{y} \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1}) \geq K^2(\boldsymbol{y}, \boldsymbol{x}_{N+1}) + \frac{r_1(\boldsymbol{y})' r_1(\boldsymbol{y})}{\lambda_1(\boldsymbol{R}) + \|r(\boldsymbol{x}_{N+1})\|} \sum_{i=1}^{N+1} (1 - \lambda_i(\boldsymbol{R}+\boldsymbol{E})\lambda_i^{-1}(\boldsymbol{R}))(\cos\theta_i)^2.$$

Using Wely's inequalities it may also be seen that $(1 - \lambda_i(\boldsymbol{R}+\boldsymbol{E})\lambda_i^{-1}(\boldsymbol{R})) \geq -\lambda_1(\boldsymbol{E})\lambda_i^{-1}(\boldsymbol{R})$, in which case

$$\sum_{i=1}^{N+1} (1 - \lambda_i(\boldsymbol{R}+\boldsymbol{E})\lambda_i^{-1}(\boldsymbol{R}))(\cos\theta_i)^2 \geq -\lambda_1(\boldsymbol{E}) \sum_{i=1}^{N+1} \lambda_i^{-1}(\boldsymbol{R})(\cos\theta_i)^2.$$

Using (5) we see that $\sum_{i=1}^{N+1} \lambda_i^{-1}(\boldsymbol{R})(\cos\theta_i)^2 = \frac{r_1(\boldsymbol{y})}{\|r_1(\boldsymbol{y})\|}' \boldsymbol{R}^{-1} \frac{r_1(\boldsymbol{y})}{\|r_1(\boldsymbol{y})\|}$. This last term is greater than the smallest eigenvalue of $\boldsymbol{R}^{-1}$ which is $\lambda_1^{-1}(\boldsymbol{R})$ (see also Bhatia, 1997, Ch. III). Therefore, we obtain

$$D(\boldsymbol{y} \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1}) \geq K^2(\boldsymbol{y}, \boldsymbol{x}_{N+1}) - \|r(\boldsymbol{x}_{N+1})\|\lambda_1^{-1}(\boldsymbol{R}) \frac{r_1(\boldsymbol{y})' r_1(\boldsymbol{y})}{\lambda_1(\boldsymbol{R}) + \|r(\boldsymbol{x}_{N+1})\|}.$$

Since $\|r(\boldsymbol{x}_{N+1})\| \frac{r_1(\boldsymbol{y})' r_1(\boldsymbol{y})}{1+\|r(\boldsymbol{x}_{N+1})\|/\lambda_1(\boldsymbol{R})} \lambda_1^{-2}(\boldsymbol{R}) \leq \|r(\boldsymbol{x}_{N+1})\| r_1(\boldsymbol{y})' r_1(\boldsymbol{y}) \lambda_1^{-2}(\boldsymbol{R})$ we have

$$D(\boldsymbol{y} \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1}) \geq K^2(\boldsymbol{y}, \boldsymbol{x}_{N+1}) - \|r(\boldsymbol{x}_{N+1})\| r_1(\boldsymbol{y})' r_1(\boldsymbol{y}) \lambda_1^{-2}(\boldsymbol{R}).$$

Since $\lambda_1(\boldsymbol{R}) = \max_{x \neq 0} \frac{x' \boldsymbol{R} x}{x' x}$ (again Bhatia, 1997, Ch. III), then, using $x = (0, 0, \ldots, 1, \ldots, 0)'$, $\lambda_1(\boldsymbol{R}) \geq \max_i \boldsymbol{R}_{ii}$. Assuming $\max_i \boldsymbol{R}_{ii} \geq 1$ we obtain

$$D(\boldsymbol{y} \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1}) \geq K^2(\boldsymbol{y}, \boldsymbol{x}_{N+1}) - \frac{\|r(\boldsymbol{x}_{N+1})\|}{(\max_i \boldsymbol{R}_{ii})^2} \left(K^2(\boldsymbol{y}, \boldsymbol{x}_{N+1}) + \sum_{i=1}^{N} K(\boldsymbol{y}, \boldsymbol{x}_i)^2 \right). \quad (6)$$

Given that $\boldsymbol{R}_{ii} = 1$, $i = 1, \ldots, N+1$, $\lambda_1(\boldsymbol{R}) \geq 1$ and we obtain

$$D(\boldsymbol{y} \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1}) \geq K^2(\boldsymbol{y}, \boldsymbol{x}_{N+1})(1 - \|r(\boldsymbol{x}_{N+1})\|) - \|r(\boldsymbol{x}_{N+1})\| \sum_{i=1}^{N} K(\boldsymbol{y}, \boldsymbol{x}_i)^2.$$

This bound has the property of being equal to $D(\boldsymbol{y} \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1})$ when $\boldsymbol{x}_{N+1}$ is not correlated with other $\boldsymbol{x}_i$ points (i.e., $\|r(\boldsymbol{x}_{N+1})\| = 0$, see (4)). However, we experimented with the above bound and obtain basically the same results when ignoring the last term. Accordingly, note that $\|r(\boldsymbol{x}_{N+1})\| \sum_{i=1}^{N} K(\boldsymbol{y}, \boldsymbol{x}_i)^2 \leq N^{\frac{3}{2}}$ and therefore

$$D(\boldsymbol{y} \mid \boldsymbol{D}_N, \boldsymbol{x}_{N+1}) \geq K^2(\boldsymbol{y}, \boldsymbol{x}_{N+1})(1 - \|r(\boldsymbol{x}_{N+1})\|) - N^{\frac{3}{2}}.$$

By ignoring the $-N^{\frac{3}{2}}$ term, that now does not depend on $\boldsymbol{x}_{N+1}$, and adding over m we obtain the score:

$$A(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N) = \frac{(1 - \|r(\boldsymbol{x}_{N+1})\|)}{m} \sum_{j=1}^{m} \frac{\sigma^4 K(\boldsymbol{y}_j, \boldsymbol{x}_{N+1})^2}{\sigma^2}. \quad (7)$$

Changing correlations to covariances (any predictive covariance structure), and identifying $\sigma^2 = \sqrt{V(\boldsymbol{y}_j)}\sqrt{V(\boldsymbol{x}_{N+1})}$, the score becomes

$$A(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N) = \frac{1 - \|r(\boldsymbol{x}_{N+1})\|}{m\sqrt{V(\boldsymbol{x}_{N+1})}} \sum_{j=1}^{m} \frac{c(\boldsymbol{y}_j, \boldsymbol{x}_{N+1})^2}{\sqrt{V(\boldsymbol{y}_j)}}, \tag{8}$$

where $\|r(\boldsymbol{x}_{N+1})\|^2 = \sum_{i=1}^{N} c(\boldsymbol{x}_{N+1}, \boldsymbol{x}_i)^2$ (and $V(\boldsymbol{y}) = c(\boldsymbol{y}, \boldsymbol{y})$, the variance at $\boldsymbol{y}$). From (4) we see that if $\|r(\boldsymbol{x}_{N+1})\| = 0$, $ALC(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N) = \frac{1}{m}\sum_{j=1}^{m} \sigma^2 K(\boldsymbol{y}_j, \boldsymbol{x}_{N+1})^2 = \frac{1}{m}\sum_{j=1}^{m} \frac{c(\boldsymbol{y}_j, \boldsymbol{x}_{N+1})^2}{\sqrt{V(\boldsymbol{x}_{N+1})V(\boldsymbol{y}_j)}} = A(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N)$. From this last identity, we see that if there is no correlation structure, $ALC(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N) = A(\boldsymbol{x}_{N+1} \mid \boldsymbol{D}_N) = \frac{1}{m}V(\boldsymbol{x}_{N+1})$.

Acknowledgments

Part of this research was completed while J. Andrés Christen was visiting the Department of AMS at UCSC with a UC-MEXUS-CONACYT sabbatical grant. The second author was partially supported by the National Science Foundation grant NSF-Geomath 0417753.

References

Bhatia, R. (1997). *Matrix Analysis*. New York: Springer.

Bursztyn, D., Steinberg, D. M. (2006). Comparison of designs for computer experiments. *J. Statist. Plann. Infer.* 136(3):1103–1119.

Christen, J. A., Fox, C. (2009). A general purpose sampling algorithm for continuous distributions (the t-walk). *Bayesian Anal.* (revision).

Cioppa, T. M., Lucas, T. W. (2007). Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics* 49(1):45–55.

Cohn, D. A. (1996). Neural network exploration using optimal experiment design. *Neur. Netw.* 9(6):1071–1083.

Fang, K. T., Li, R. Z. (2006). Uniform design for computer experiments and its optimal properties. *Int. J. Mat. Product Technol.* 25(1–3):198–210.

Feynman, R. P. (1985). *"Surely You are Joking Mr. Feynman!" Adventures of a Curious Character*. New York: W. W. Norton & Company.

Forest, C. E., Stone, P. H., Sokolov, A. P. (2006). Estimated PDFs of climate system properties including natural and anthropogenic forcings. *Geophys. Res. Lett.* 33(1):L01705.

Gramacy, R. (2005). Bayesian Treed Gaussian Process Models. Ph.D. thesis, University of California at Santa Cruz, Santa Cruz, CA.

Gramacy, R., Lee, H. (2008). Bayesian treed Gaussian process models with an application to computer modeling. *J. Amer. Statist. Assoc.* 103:1119–1130.

Gramacy, R., Lee, H. (2009). Adaptive design and analysis of supercomputer experiments. *Technometrics* 51:130–145.

Kennedy, M. C., O'Hagan, A. (2001). Bayesian calibration of computer models. *J. Roy. Statist. Soc. Ser. B* 63:425–464.

Lehman, J. S., Santner, T. J., Notz, W. I. (2004). Designing computer experiments to determine robust control variables. *Statistica Scinica* 14(2):571–590.

Mease, D., Bingham, D. (2006). Latin hyperrectangle sampling for computer experiments. *Technometrics* 48(4):467–477.

Sansó, B., Forest, C., Zantedeschi, D. (2008). Inferring climate system properties using a computer model (with discussion). *Bayesian Anal.* 3:1–62.

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

Schonlau, M., Welch, W. J., Jones, D. R. (1998). Global versus local search in constrained optimization of computer models. In: Flournoy, N., Rosenberger, W. F., Wong, W. K., eds. *New Developments and Applications in Experimental Design (IMS Lecture Notes Monograph Series, Volume 34)*. Institute of Mathematical Statistics, pp. 11–25.

Sokolov, A. P., Stone, P. H. (1998). A flexible climate model for use in integrated assessments. *Climate Dyn.* 14:291–303.

Steinberg, D. M., Bursztyn, D. (2004). Data analytic tools for understanding random field regression models. *Technometrics* 46(4):411–420.

Stinstra, E., Den Hertog, D., Stehouwer, P., Vestjens, A. (2003). Constrained maximin designs for computer experiments. *Technometrics* 45(4):340–346.

Williams, B. J., Santner, T. J., Notz, W. I. (2000). Sequential designs of computer experiments to minimize integrated response functions. *Statistica Scinica* 10:1133–1152.