ADAPTIVE SAMPLING WITH TOPOLOGICAL SCORES

Dan Maljovec,[1] *Bei Wang,*[1,*] *Ana Kupresanin,*[2] *Gardar Johannesson,*[2] *Valerio Pascucci,*[1] *& Peer-Timo Bremer*[1,2]

[1]*Scientific Computing and Imaging Institute, University of Utah, 72 South Central Campus Drive, Salt Lake City, UT 84112*

[2]*Lawrence Livermore National Laboratory, 7000 East Avenue, Livermore, CA 94550-9234*

Original Manuscript Submitted: 07/18/2012; Final Draft Received: 07/18/2012

Understanding and describing expensive black box functions such as physical simulations is a common problem in many application areas. One example is the recent interest in uncertainty quantification with the goal of discovering the relationship between a potentially large number of input parameters and the output of a simulation. Typically, the simulation of interest is expensive to evaluate and thus the sampling of the parameter space is necessarily small. As a result choosing a "good" set of samples at which to evaluate is crucial to glean as much information as possible from the fewest samples. While space-filling sampling designs such as Latin Hypercubes provide a good initial cover of the entire domain, more detailed studies typically rely on adaptive sampling: Given an initial set of samples, these techniques construct a surrogate model and use it to evaluate a scoring function which aims to predict the expected gain from evaluating a potential new sample. There exist a large number of different surrogate models as well as different scoring functions each with their own advantages and disadvantages. In this paper we present an extensive comparative study of adaptive sampling using four popular regression models combined with six traditional scoring functions compared against a space filling design. Furthermore, for a single high-dimensional output function, we introduce a new class of scoring functions based on global topological rather than local geometric information. The new scoring functions are competitive in terms of the root mean square prediction error but are expected to better recover the global topological structure. Our experiments suggest that the most common point of failure of adaptive sampling schemes are ill-suited regression models. Nevertheless, even given well-fitted surrogate models many scoring functions fail to outperform a space-filling design.

KEY WORDS: *adaptive sampling, experimental design, topological techniques*

1. INTRODUCTION

As the accuracy and availability of computer simulations improves, their results are increasingly used to inform far reaching decisions. Experts in areas ranging from building and automobile design to national energy policy regularly use predictive simulations to evaluate alternative approaches. However, virtually all simulations are based on approximate models and an incomplete knowledge of the underlying physics and thus do not accurately predict the phenomena of interest. Furthermore, there typically do exist a large number of parameters, e.g. material properties, boundary conditions, sub-scale parameters, etc., that influence the outcome and are used to tune the simulation to match experiments or observations. Nevertheless, usually no perfect set of parameters exists nor is this type of fitting possible for the most interesting case of truly predictive simulations, e.g. weather or climate forecasts. In such scenarios the simulation parameters represent a significant source of uncertainty and a single best guess even by an expert is not very reliable. Consequently, understanding the uncertainty involved in a prediction, the range of possible outcomes, and the confidence in the results is of interest.

One common approach to address these issues is to create not one but an ensemble of simulations each with slightly different parameter settings. The resulting collection of outcomes is then analyzed to determine the likelihood of various scenarios to occur and to assess the confidence in the prediction. The challenge lies in the fact that the

[*] Correspond to: Bei Wang, E-mail: beiwang@sci.utah.edu, URL: http://www.sci.utah.edu/∼beiwang/

dimension of the parameter space can be large, e.g. tens or hundreds of parameters, and each simulation might be expensive, e.g. taking hundreds or thousands of CPU hours. Therefore, the space of possible solutions can only be sampled very sparsely and each simulation must be carefully chosen to provide the maximal amount of information. A first order solution is to sample the parameter space as evenly as possible and there exist a number of approaches such as the Latin Hypercube design [1], orthogonal arrays [2], and related techniques [3] that address this issue.

However, in practice much of the parameter space might be invalid, producing clearly unphysical results, or simply uninteresting and easy to predict. In these situations a space-filling sampling would waste a large amount of resources. Instead, adaptive sampling is used to iteratively guide the choice of future computer runs by repeatedly analyzing the model based on the existing samples. The most common approach to do adaptive sampling is based on the use of statistical prediction models (also known as surrogate models, response functions, statistical emulators, meta models) such as Gaussian processes models (GPMs) [4] or multivariate adaptive regression splines (MARS) [5]. The basic concept of adaptive sampling is relatively simple and well established: First, one constructs a prediction model based on an initial set of samples (training points); Second, a large set of candidate points is chosen in the parameter space and the prediction model is evaluated at these points; Third, each candidate point is assigned a *score* based on, for example the estimated prediction error; Finally, the candidate(s) with the highest score are selected and evaluated by running the corresponding simulation.

While the basic pipeline of adaptive sampling is universally accepted, combining different regression models with different scoring functions often leads to drastically different results. Here, we present an extensive experimental study combining four popular statistical models, a GPM [4], two implementations of MARS [5] (available in the `mda` and the `earth` libraries of the statistical programming language `R`), and a neural network (NNET) [6] (from the `nnet` library in `R`), with six traditional as well as three new topology based scoring functions. Using a variety of test functions in two to five dimensions each combination is compared against a Latin Hypercube design of the same sample size to evaluate the potential advantage of adaptive sampling in general. As discussed in more detail in Section 4, the most dominant factor in the results is the choice of the statistical prediction model as some models appear to be ill-suited for several test functions and for all scoring functions they produce unsatisfactory results. Given an appropriate prediction model some trends appear that favor information theoretic scoring functions even though for several experiments all scoring functions fail to out-perform the space filling design.

In addition to the experimental results we also introduce a new class of scoring functions based on global topological rather than local geometric information. In particular, we define three new scoring functions primarily aimed at recovering the global structure of a function. Nevertheless, our results show that the new scoring functions remain competitive in terms of mean square prediction error.

2. BACKGROUND AND RELATED WORK

Here we introduce some of the necessary background and discuss related work in both adaptive sampling as well as topological analysis and visualization.

2.1 Statistical Prediction Models

Since computer simulations are generally computationally expensive, a standard approach to analyze them is to build a statistical prediction model (PM), and use it in place of the actual simulation code in further analysis, to guide the selection of upcoming ensemble of simulations. We now give a brief overview of some commonly used statistical PMs; see Fang, Li, and Sudjianto [7] for further details.

Let $y(\mathbf{x})$ denote the output of the simulation code, where the vector input variable, $\mathbf{x}$, is restricted to the d-dimensional unit cube $[0,1]^d$. This assumption is easily relaxed to any “rectangular” input space. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in [0,1]^d$ be an ensemble of n input vectors and denote the resulting simulation data of interest by $y_i = y(\mathbf{x}_i)$, $i = 1, \ldots, n$.

Gaussian Process Model (GPM). One of the statistical models most frequently used for prediction is the Gaussian process model (GPM), first used in the context of computer experiments by Sacks [8] in 1989. A common practice is to place a homogeneous GP prior on the family of possible output functions. Then, the predictor is given by the posterior mean conditional on the output of the computer experiment.

The GPM places a prior on the class of possible functions $y(\mathbf{x})$. We denote by $Y(\mathbf{x})$ the random function whose distribution is determined by the prior. Suppose that $Y(\mathbf{x}) = \mu + Z(\mathbf{x})$, where μ is a mean parameter and $Z(\mathbf{x})$ is a Gaussian stochastic process with mean 0, constant variance σ^2, and an assumpted (parametric) correlation function. An example of popular correlation model is the power exponential correlation function, given by $R(\mathbf{x}, \mathbf{x}') = exp(-h(\mathbf{x}, \mathbf{x}'))$, where $h(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{d} \theta_j |x_j - x'_j|^{p_j}$ with $\theta_j \geq 0$ and $1 \leq p_j \leq 2$. Here, $\mu, \sigma^2, \theta_j, p_j$ are the parameters of the prior model. The values of these parameters are estimated using the ensemble data, $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$, typically using a likelihood-based objective function. In situations where the output is quite smooth, it is often the case that $p_j = 2$ holds for all j, resulting in a Gaussian correlation function.

The predictor $\hat{Y}(\mathbf{x})$ for $Y(\mathbf{x})$ is the posterior mean of $Y(\mathbf{x})$ given $\{(\mathbf{x}_i, y_i)\}$ and is available in a closed form expression. Similarly, the mean squared prediction error (MSPE) of $\hat{Y}(\mathbf{x})$ (the prediction error variance), taking into account the uncertainty from estimating μ by maximum likelihood (but not estimating the correlation parameters), is also available in a closed form. For more details, see Rasmussen [4].

Multivariate Adaptive Regression Splines (MARS). Another popular class of PMs are Friedman's multivariate adaptive regression splines [5] (see also Hastie, Tibshirani, and Friedman [9] for a gentle introduction).

MARS is a non-parametric regression method that uses a collection of simple basis functions to build a complex and flexible response function. At the core of MARS are piecewise linear basis functions, given by $\max(0, x - k)$ and $\max(0, k - x)$, where k is the knot (the break point). These simple functions are used to build a collection of basis functions, $\{\max(0, x_j - k), \max(0, k - x_j)\}$, where $k \in \{x_{1j}, \ldots, x_{nj})$ for $j = 1, \ldots, d$. If all the input values are different in the training data this will yield $2nd$ basis functions. The MARS predictor is then given by $\hat{y}(\mathbf{x}) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(\mathbf{x})$, where the β_m's are coefficients and the h_m's are basis functions, which are either given by a single function from the collection of simple basis functions or a product of two or more of such functions.

For a given set of basis functions, $h_1, \ldots, h_m$, the β coefficients are estimated using least squares. The main novelty of the MARS is how the basis functions are selected using a forward selection process, followed by a backward elimination process. The forward pass starts with a model that only includes the constant term β_0 and then adds simple basis functions in pairs in a greedy fashion. This yields a model that over fits the training data (i.e. has too many terms). The backward elimination process drops terms using the generalized cross-validation (GCV) criterion, which compromises between the fidelity of the model and its complexity (size).

As with most non-parametric regression methods, there is no direct method to assess their prediction error. However, the prediction error can be estimated by bootstrap methods [10]. In short, an ensemble of MARS models is created by repeatedly resampling, with replacement, the original training data $\{(\mathbf{x}_i, y_i)\}$ and a MARS model is fitted to each batch of resampled data. An estimate of the prediction error is then given by the standard deviation of the predictions provided by the bootstrap ensemble.

Neural Networks (NNET). Another popular class of PMs is given by neural networks (NNETs), in particular by simple feed-forward neural networks with a single hidden layer (see e.g. Ripley [6] and Hastie et al. [9] for further details).

The single layer NNET is simply a non-linear statistical model, where the response of interest is modeled as a linear combination of M derived features (the hidden units of the NNET); $\hat{y}(\mathbf{x}) = \beta_0 + \beta^T \mathbf{z}$, $\mathbf{z} = (z_1, \ldots, z_M)^T$, where $z_m = \sigma(\alpha_{0m} + \alpha_m^T \mathbf{x})$. The activation function, $\sigma(v)$ is usually taken as the sigmoid function, $\sigma(v) = 1/(1 + e^{-v})$.

The unknown parameters of the NNET, which we denote by θ and are often referred to as weights, and consist of the $M(d+1)$ α's of the hidden layer and the $M+1$ β's for the response model. The weights are estimated by minimizing the sum-of-squares, $R(\theta) = \sum_{i=1}^{n} (y_i - \hat{y}(\mathbf{x}_i))^2$. However, for a large NNET, the resulting NNET overfits the data (i.e., the NNET fits the training data well, but performs poorly on independent validation data). A common solution is to add a regularization term that shrinks the weights to zero, such as $J(\theta) = \sum_{m=0}^{M} \beta_m^2 + \sum_{m=0,j=1}^{M,d} \alpha_{mj}^2$, and then minimize $R(\theta) + \lambda J(\theta)$. The minimization can be carried out in an efficient manner using gradient-based methods, as the gradient of the NNET can be computed easily using the chain rule for differentiation. As in the case of MARS, the prediction error of the NNET can be estimated using bootstrap methods.

2.2 Design of Computer Experiments

Experimental designs relevant to computer experiments (i.e. sampling) are often broadly categorized into two classes: space-filling and criterion-based designs. Intuitively, it is natural to consider a space-filling design strategy to minimize the overall prediction error and a number of approaches have been studied. These include methods based on selecting random samples e.g. Latin hypercube designs, distance-based designs and uniform designs. A thorough discussions of the various strategies may be found in Satner et al. [11], Koehler and Owen [12], and Bates et al. [13]. Intuitively, the goal is to ensure that the input points are uniformly distributed over the range of each input dimension. In our setting, space-filling designs are attractive for an initial exploratory analysis. However, their applicability for detailed studies is limited by their very construction rationale, that is, the assumption that the features of the response model are equally likely to be found anywhere in the input space. More specifically, following a space-filling design, we will have no freedom to adapt our selection of input points to information gathered as simulations are completed. Instead, using the knowledge contained in a partially constructed model may allows us to adaptively select the samples in regions of interest. If samples are chosen correctly, this strategy will greatly improve prediction accuracy and efficiency compared to a pure space-filling design.

This leads to the second class of experimental designs: those constructed by adding one or several points at a time to the initial model. New points are selected from a large candidate set based on various statistical criteria and the prediction of the existing (partial) model. Due to their iterative nature these designs are usually referred to as sequential, or adaptive. Sequential designs based on optimizing statistical criteria such as mean squared prediction error or the notion of entropy have also been used to construct designs for computer experiments [11]. Here we use a simple sampling of the range space, see Section 3.1, as well as the mean squared prediction error (MSPE) and the expected improvement as criteria.

Maximum mean squared prediction error. The mean square prediction error (MSPE) is simply the prediction error of PM at a given new input point. In the case of the GPM, the prediction error is available in a closed form, but is estimated using bootstrap for MARS and NNET. The MSPE criterion aims at selecting the point from the candidate set that has the largest MSPE.

In the case of a stationary GPM with a constant mean, this results in criterion that spreads points out, but at different density along each axis, and typically starts out by populating points near the boundary of the input space. In the case of MARS and NNET, which can capture very non-stationary behavior, the criterion typically populates points in the region of the input space which is most sensitive to bootstrap resampling, that is, where there is a large variation in the response.

Maximum expected improvement (EI). The expected improvement (EI) criterion was proposed by [14] and originally developed in the context of global optimization [15]. Lam [16] considered a modification of this criterion with the goal of obtaining a good global fit of the GPM instead of locating the global optimum or optima. Intuitively, the objective here is to search for "informative" regions in the domain that will help improve the global fit of the model, where "informative" means regions with significant variation in the response variable.

In case of the GPM, for each potential input point $\mathbf{x}$, Lam defined the improvement as $I(\mathbf{x}) = (Y(\mathbf{x}) - y(\mathbf{x}_{j^*}))^2$, where $y(\mathbf{x}_{i^*})$ is the observed output at the sampled point $\mathbf{x}_{i^*}$ closest (in Euclidean distance) to the candidate point $\mathbf{x}$. The maximum expected improvement criterion advises to select as the next point the one that maximizes the expected improvement $EI(\mathbf{x}) = (\hat{Y}(\mathbf{x}) - y(\mathbf{x}_{i^*}))^2 + \hat{\sigma}^2(\mathbf{x})$. One typically works with the square-root of the $E(I)$, which is at the same scale as response. For the details of the derivation of $E(I)$ using a GPM, we refer to [16]. The EI criterion can be extended to MARS and NNET by simply replacing $\hat{Y}(\mathbf{x})$ with the bootstrap average and the prediction variance $\hat{\sigma}^2(\mathbf{x})$ with the bootstrap variance.

The estimate of expected improvement uses two search components, one local and one global. The first (local) component of the expected improvement will tend to be large at points where the increase in response over the nearest sampled points is large. The second (global) component is large at points with the largest prediction error.

2.3 Morse-Smale Complex and Its Approximation

To quantify the expected topological impact of a point during adaptive sampling, we introduce a key topological structure, the *Morse-Smale Complex*, which forms the basis for the new scoring functions introduced in Section 3.2.

Topological structures, such as contour trees [17], Reeb graphs [17–20] and Morse-Smale complexes [21, 22] provide abstract representations for scalar functions. These structures can be used to define a wide variety of features in various applications, ranging from medical [23], to physics [24, 25] and material science [26]. To analyze and visualize high-dimensional data, several topological approaches have been proposed, in particular, [27, 28].

Let $\mathbb{M}$ be a smooth manifold without boundary and $f : \mathbb{M} \to \mathbb{R}$ be a smooth function with gradient ∇f. A point $x \in \mathbb{M}$ is called *critical* if $\nabla f(x) = 0$, otherwise it is *regular*. If the Hessian matrix at a critical point is non-singular then the critical point is called *non-degenerate*. At any regular point x the gradient is well-defined and integrating it in both directions traces out an integral line, $\gamma : \mathbb{R} \to \mathbb{M}$, which is a maximal path whose tangent vectors agree with the gradient [21]. Each integral line begins and ends at critical points of f. The *ascending/descending manifolds* of a critical point p is defined as all the points whose integral lines start/end at p. The descending manifolds form a complex called a *Morse complex* of f and the ascending manifolds define the Morse complex of $-f$. The set of intersections of ascending and descending manifolds creates the *Morse-Smale* complex of f. Each cell (*crystal*) of the Morse-Smale complex is a union of integral lines that all share the same origin and the same destination. In other words, all the points inside a single crystal have uniform gradient flow behavior. These crystals yield a decomposition into monotonic, non-overlapping regions of the domain, as shown in Fig. 1(a)-(c) for a two-dimensional height function.

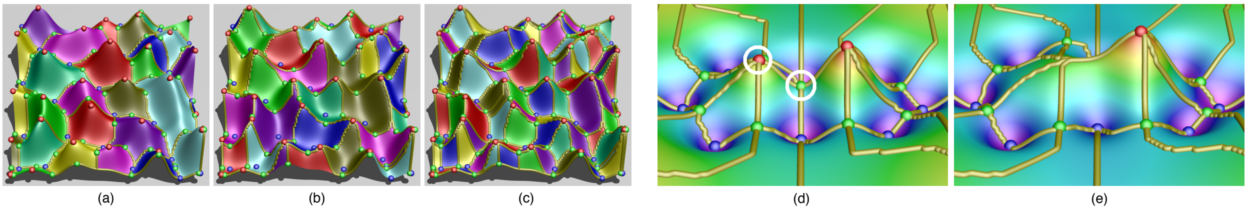


FIG. 1: Left: (a) Ascending manifolds, (b) descending manifolds and (c) Morse-Smale complex. Right: A 2D Morse-Smale complex before (c) and after (d) persistence simplification. Maxima are red, minima are blue and saddles are green.

In two and three dimensions, discrete Morse-Smale complex on piecewise linear functions can be constructed [17, 21, 22, 29]. For high dimensional point cloud data, the Morse-Smale complex can only be approximated [30, 31]. Here, we give an overview of the approximation approach in high dimension detailed in [31, 32], which is crucial in our algorithm pipeline. First, the domain is approximated by a k nearest neighbor (kNN) graph. The algorithm uses a discrete approximation of the integral line by following the paths of steepest accent and descent among neighboring points in the graph, based on a quick-shift algorithm [33]. The neighbors of a point x_i is defined as $\mathrm{adj}\,(x_i) = \{x_j \mid x_j \in \mathrm{knn}\,(x_i)$ or $x_i \in \mathrm{knn}\,(x_j)\}$. Its steepest ascent is $\arg\max_{x_j \in \mathrm{adj}\,(x)} ||f(x_j) - f(x_i)||/||x_i - x_j||$, while its steepest descent is $\arg\max_{x_j \in \mathrm{adj}\,(x)} ||f(x_i) - f(x_j)||/||x_i - x_j||$. Each point x_i is then assigned to a crystal of the Morse-Smale complex which is a union of approximated integral lines that all share the same origin and the same destination. The domain is then partitioned into regions $\{C_1, C_2, ..., C_l\}$ where $\bigcup_i C_i = \{x_i\}_{i=1}^n$. In this approximated Morse-Smale complex, a maximum/minimum has no ascending/descending neighbors, respectively.

2.4 Persistence

One advantage of the Morse-Smale complex is that it can be used to associate a notion of *significance* to the critical points. For example, as shown in Fig. 1(d), the left peak (the circled red maxima) is considered less important topologically than its nearby peak (un-circled red maxima to the right) as it is lower. Therefore, at certain scale, we would like to represent this feature as a single peak instead of two separate peaks, as shown in Fig. 1(e). This simplification procedure and the notion of *scale* is defined through the concept of *persistence*.

The theory of persistence was first introduced in [34, 35], but borrows from the conventional notion of the saliency of watersheds in image segmentation. It has since been applied to a number of problems, including sensor networks [36], surface description and reconstruction [37], protein shapes [38], images analysis [39], and topological de-noising [40]. In visualization, it has been used to simplify Morse-Smale complexes [41, 42], Reeb graphs [43] and contour trees [23]. Here we introduce persistence for a 1D (single variable) function [38] and refer to [34, 35, 44] for its general settings.

For a one-dimensional smooth function $f : \mathbb{R} \to \mathbb{R}$, persistence can be described through the number of connected components in the sublevel sets, and by tracking the birth and death of these components. In particular, components are created and destroyed only at sublevel sets containing critical points. Pairing the critical point that creates a component with the one that destroys it thus creates a pairing of critical points. Suppose f has non-degenerate critical points with distinct function values. We have two types of critical points, (local) maxima and (local) minima. We consider the sublevel sets of f, $F_t = f^{-1}(-\infty, t]$ and track the connectivity of F_t as we increase t from $-\infty$. As shown in Fig. 2(a), when we pass the minimum point a, a new component appears in the sublevel sets, which is represented by the minimum a, with a *birth* time $f(a)$. Similarly when we pass the minimum b and c, two new components are born with birth time $f(b)$ and $f(c)$, respectively. When we pass the maximum d, two components represented by a and c are merged and the maxima is paired with the younger (higher) of the two minimum that represent the two components, that is, d and c are paired, where $f(d)$ is the *death* time of the component represented by c. We define the *persistence* of the pair to be $f(d) - f(c)$, which corresponds to the significant of a topological feature. We then encode persistence in the *persistence diagram*, $\mathrm{Dgm}(f)$, by mapping the critical point pair to a point $(f(c), f(d))$ on the 2D plane. Similarly we pair e with b and f with a, resulting two more points in $\mathrm{Dgm}(f)$. For technical reasons, the diagonal is considered as part of the persistence diagram that contains an infinite number of points.

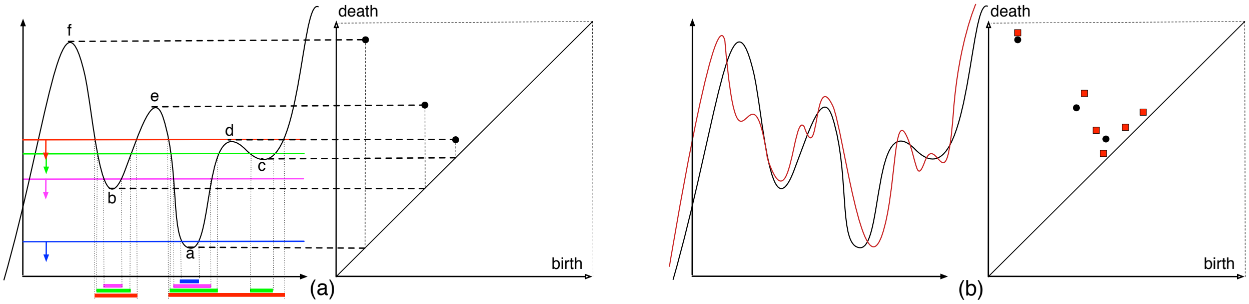


FIG. 2: (a) A 1D function with three local minima and three local maxima. The critical points are paired, and each pair is encoded as a point in the persistence diagram on the right. (b) Left, two functions $f, g : \mathbb{R} \to \mathbb{R}$ with small L_∞-distance. Right, their corresponding persistence diagrams $\mathrm{Dgm}(f)$ (circles) and $\mathrm{Dgm}(g)$ (squares) have small bottleneck distance.

Recent results show that persistence diagrams are stable under small perturbations of the functions [44, 45]. Let $p = (p_1, p_2), q = (q_1, q_2)$ be two points in the persistence diagram, let $||p - q||_\infty = \max\{|p_1 - q_1|, |p_2 - q_2|\}$. For functions $f, g : \mathbb{R} \to \mathbb{R}$, $||f - g||_\infty = \sup_x |f(x) - g(x)|$. The *bottleneck distance* between two multi-sets of points in $\mathrm{Dgm}(f)$ and $\mathrm{Dgm}(g)$ is $d_B(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) = \inf_\gamma \sup_x ||x - \gamma(x)||_\infty$, where $x \in \mathrm{Dgm}(f)$ and $y \in \mathrm{Dgm}(g)$ range over all points, and γ ranges over all bijections from $\mathrm{Dgm}(f)$ to $\mathrm{Dgm}(g)$ [45]. The Stability Theorem states that the persistence diagrams satisfy: $d_B(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) \leq ||f - g||_\infty$. This is illustrated in Fig. 2(b).

Using the Morse-Smale complex the persistence pairing can be created by successively canceling the two critical points connected in the complex with minimal persistence while avoiding certain degenerate situations. This assigns a persistence to each critical point in the complex which, intuitively, describes the scale at which a critical point would disappear through simplification. Note that in the approximate Morse-Smale complexes created from high dimensional point clouds [32] only a subset of theoretically possible cancellations can be performed which changes the persistences slightly. However, we have not observed any negative effects of the approximation.

3. ADAPTIVE SAMPLING

Two main families of prediction models (PMs) are used in the uncertainty quantification (UQ): regression models such as MARS and stochastic models such as Gaussian processes. Our general pipeline as illustrated in Fig. 3 is applicable to both families.

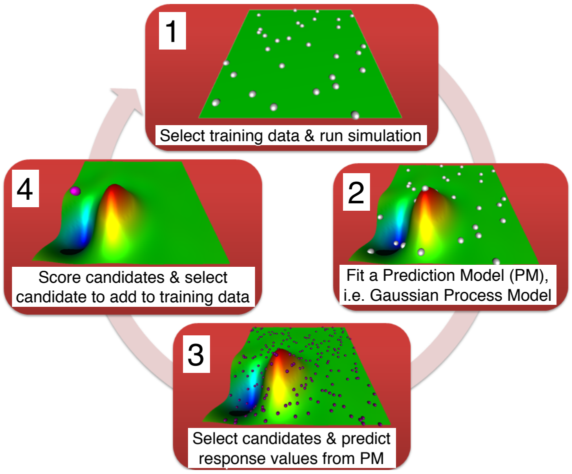

1
Select training data & run simulation
2
Fit a Prediction Model (PM), i.e. Gaussian Process Model
3
Select candidates & predict response values from PM
4
Score candidates & select candidate to add to training data

FIG. 3: The iterative pipeline for adaptive sampling. Starting with a set of training points (top) a prediction model is create (right). The model is evaluated at a large number of candidate points (bottom) typically created through a space-filling design. Each candidate point is assigned a score indicating the expected information gain were this point being evaluated. Finally, the point with the highest score is selected and evaluated using the simulation (left). Finally, the new sample is added to the training data and the process is repeated.

We begin by selecting some initial training data, running the simulation and obtaining a collection of true responses at these data points. Second, we fit a prediction model (PM), i.e. a Gaussian Process Model, from the initial set of training data. Third, a large set of candidate points is chosen in the parameter space using Latin Hypercube Sampling (LHS), and the PM is evaluated at these points. It is important to note that we use PM to approximate values at these candidate points, which is highly efficient. Fourth, each candidate point is assigned a score based on some adaptive sampling score functions. Finally, the candidates with the highest scores are selected and added to the set of training data to begin a new cycle.

Traditional score functions are either based on point density, where candidate points which are further away from the existing training data get higher scores; or are based on prediction accuracy, where candidates with higher prediction uncertainty get higher scores. We propose a third class of score functions based on topological information, where candidate points in the area of larger topological changes are assigned higher scores.

3.1 Traditional Scoring Functions

We first review several traditional scoring functions. We compare our topological scoring functions to these metrics. There are three main scoring functions, namely, Delta, ALM and EI, as detailed below. Each of them is further augmented with a distance penalization factor, creating three additional scoring functions.

Let $\mathcal{T} = \{\mathbf{z}\}_{i=1}^{n}$ be the set of n training points in dimension d. The true response at a point $\mathbf{z} \in \mathcal{T}$ is denoted as $y(\mathbf{z})$. Let $\mathcal{S} = \{\mathbf{x}\}_{i=1}^{m}$ be the set of m candidate points in dimension d. The predicted response at a point $\mathbf{x} \in \mathcal{S}$ is denoted as $\hat{y}(\mathbf{x})$.

Delta. This criterion can be seen as a way to evenly sample the range space of the function. It is defined as the absolute value of the difference between the predicted response at a candidate point and the response at the nearest training point. Points are chosen wherever the response model predicts a large gap in function value. Note that while the delta criterion is very intuitive it does not consider either gradient magnitudes nor “predictability”. For example, a point in the middle of a steep but linear ramp is easily predicted even though it may have a large difference in function value. Similarly, a point with a large difference in function value far away from the nearest sample may not be as interesting as a slightly smaller difference in a highly sampled region. Formally, for a point $\mathbf{x} \in \mathcal{S}$, $Delta(\mathbf{x})$ is the absolute difference, to the q-th power, between the predicted response and the response observed at the closest point in the training sample, as measured by the L_p distance metric, that is, $d(\mathbf{x}, \mathbf{x}') = (\sum_{i=1}^{d} |x_i - x'_i|^p)^{1/p}$. That is, for some fixed parameters p and q, let $\mathbf{x}^* = \arg\min_{\mathbf{z} \in \mathcal{T}} d(\mathbf{x}, \mathbf{z})$, then $Delta(\mathbf{x}) = |\hat{y}(\mathbf{x}) - y(\mathbf{x}^*)|^q$. In the default setting, $p = 2$ and $q = 2$.

ALM. This is the Active Learning MacKay criterion described in [46] which attempts to optimize the predictive variance. The idea is that the variance represents a notion of uncertainty in the prediction and new samples should be evaluated in the least well understand regions of the parameter space. For the GPM the variance can be computed directly from the model which appears to be a significant advantage (see Section 4). For the other prediction models we use bootstrapping (as described in Section 2.1 and APPENDIX B) to estimate the variance.

EI. This is the expected improvement criterion. As discussed in Section 2.2, this can be seen as a combination of the expected prediction error used in the ALM method and the Delta criterion. Points are chosen that either show a large uncertainty in their current prediction or have a large discrepancy with the closest existing sample. Our predication model uses $EI(\mathbf{x}) = (|\hat{y}(\mathbf{x}) - y(\mathbf{x}^*)|^2 + ALM(\mathbf{x}))^{1/2}$.

Distance Penalization. Each of the above three scoring functions can be augmented with a distance penalization factor, therefore creating three additional scoring functions, namely, *DeltaDP*, *ALMDP* and *EIDP*. For DeltaDP, the Delta criterion with an additional penalty term, we can prevent samples from lying too close to the training set. The scaling attempts to balance the goal of sampling in areas of large function variance with the ability to detect yet unknown features by preferring under-sampled areas. For a point $\mathbf{x} \in \mathcal{S}$, $DeltaDP(\mathbf{x}) = Delta(\mathbf{x}) * \rho_{\mathbf{x}}$, where $\rho_{\mathbf{x}}$ is the distance scaling factor. Recall $d_{\mathbf{x}} = d(\mathbf{x}, \mathbf{x}^*)$ is the distance from $\mathbf{x}$ to the closest point in the training data, and D is a distance vector of $d_{\mathbf{x}}$ for all $\mathbf{x} \in \mathcal{S}$. $\rho_{\mathbf{x}} = \rho_{\mathbf{x}}(d_{\mathbf{x}}, d_0, p_0)$, where d_0 is the range and q_0 is the quantile (by default, d_0 is the q_0 quantile of D). If $d_{\mathbf{x}} > d_0$, set $\rho_{\mathbf{x}} = 1$, otherwise $\rho_{\mathbf{x}} = 1.5d_{\mathbf{x}} - 0.5d_{\mathbf{x}}^3$, where the coefficients are taken from spherical semivariogram. Similarly we define $ALMDP(\mathbf{x}) = ALM(\mathbf{x}) * \rho_{\mathbf{x}}$, where we approach a more space-filling point selection; and $EIDP(\mathbf{x}) = EI(\mathbf{x}) * \rho_{\mathbf{x}}$. By default we use $q_0 = 0.5$.

3.2 Topological Scoring Functions

All of the scoring functions discussed above pick sample points more or less directly based on the idea of globally improving the prediction accuracy. However, these points are not necessarily the optimal candidates. Imagine, for example, a steep mountain that (by random chance) has already been sampled both close to its peak as well as somewhere near the base. For points on the slope of the mountain, the prediction will show a large difference in function value thus making it attractive for most standard techniques. However, evaluating the prediction in more detail would also show that even taking a sizable prediction error into account the global structure of the mountain would not change by adding a point on its slope. More specifically, the single mountain would remain a single mountain for a wide range of potential new values even considering errors and uncertainty. This rationale leads to topology based scoring function aimed at discovering the global structure – the topology – of a function rather than its detailed geometry. In particular, we propose three different topology based scoring functions, named, *TopoHP*, *TopoP* and *TopoB* as detailed below, illustrated in Fig. 4.

TopoHP. The first strategy is aimed at sampling at or near predicted critical points with significant influence on the topology. It is defined as the persistence of a candidate point within an (approximated) Morse-Smale complex constructed from oversampling the current response model. Given the current response model, we evaluate its prediction at all candidate points and compute the Morse-Smale complex of the resulting point set by combining both training points and candidate points. We then assign all critical points of the complex that are part of the candidate sets their persistence as score and assign a zero score to all regular points within the candidate sets. Referring to Fig. 4(a), the silver points illustrate the training points $\mathcal{T}$ and the purple points correspond to the candidate points $\mathcal{S}$. We construct a Morse-Smale complex over $\mathcal{T} \cup \mathcal{S}$, and return the persistence of the critical points within the candidates. Here, point $\mathbf{x}$ is selected with the highest persistence, therefore, the highest $TopoHP(\mathbf{x})$.

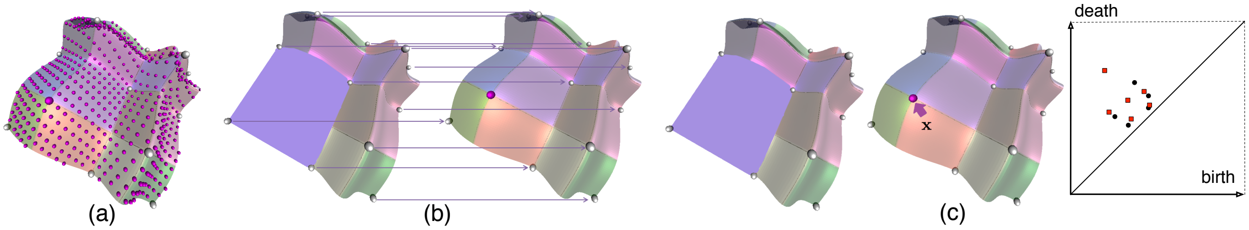


FIG. 4: (a) TopoHP: Construct a Morse-Smale complex from training data (silver) as well as all candidates with predicted responses (purple), return the persistence of the critical points within the candidates. Point $\mathbf{x}$ is selected with the highest $TopoHP(\mathbf{x})$. (b) TopoP: average change in persistence for all extrema before (left) and after (right) inserting a candidate $\mathbf{x}$ into the Morse-Smale complex. (c) Morse-Smale complexes before (left) and after (middle) inserting a candidate point $\mathbf{x}$; TopoB (right): bottleneck distance between the corresponding persistence diagrams before (circles) and after (squares) insertion.

TopoP. Similar to a bootstrapping approach, this strategy aims to evaluate how much the topology (as represented by the persistences) would change if a new candidate point is added. It is defined as the average change of persistence for all current extrema when a given candidate point with its predicted response is inserted into the Morse-Smale complex. As shown in Fig. 4(b), we first construct the Morse-Smale complex of all training data $\mathcal{T}$ (silver points). Then for each candidate point $\mathbf{x} \in \mathcal{S}$, we construct a new Morse-Smale complex consisting of $\mathcal{T} \cup \mathbf{x}$ (Fig. 4(b) right). To score the candidate $\mathbf{x}$, we compute the change in persistence for each training point $\mathbf{x} \in \mathcal{T}$ that remains as an extrema point between the original and enhanced Morse-Smale complex, and average these changes to obtain a single, nonnegative value.

TopoB. For each point $\mathbf{x} \in \mathcal{S}$, $TopoB(\mathbf{x})$ is defined as the bottleneck distance between the persistence diagram of the Morse-Smale complex over $\mathcal{T}$ versus the Morse-Smale complex consisting of $\mathcal{T} \cup \mathbf{x}$. This strategy is similar to the TopoP scoring except that the bottleneck distance not only takes the persistence values into account but also the order and nesting of the corresponding simplification. This is shown in Fig. 4(c), where left and middle illustrate the (approximated) Morse-Smale complexes before and after inserting a candidate point $\mathbf{x}$, and right displays the corresponding persistence diagrams of these complexes. $TopoB(\mathbf{x})$ is defined as the bottleneck distance between them.

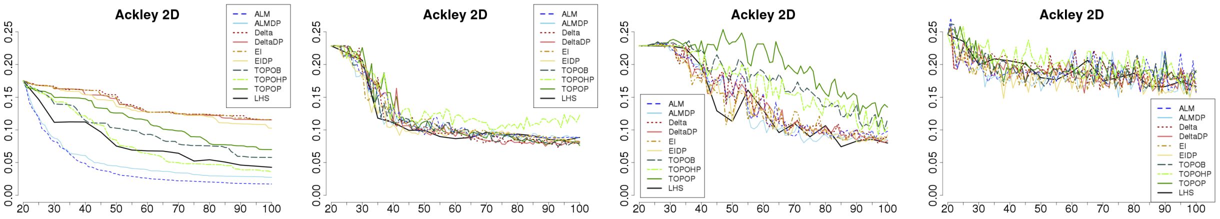
4. EXPERIMENTS

This section summarizes the different experiments and highlights apparent trends and interesting behaviors.

Example Data Sets. To evaluate the different scoring functions and understand the behavior in different scenarios we have conducted a series of experiments with well-known analytic functions, which can be generalized to high-dimensions. For example, a widely used multimodal test function from the optimization literature *Ackley* [47], as well as easily controlled test functions such as *Gaussian Mixtures*, and the *Diagonal* function. The Diagonal function consists of a sin curve aligned with the main diagonal of the unit (hyper-)cube convolved with a Gaussian kernel in the hyperplane orthogonal to the diagonal (see [31]). The Diagonal function is attractive for testing as it is not axis aligned, its topological structure is well understood and can be computed analytically, and its complexity is easily controlled. All functions with their closed forms and their 2D contour plots are shown in APPENDIX A.

Plot and Specifications. All graphs show the root mean squared error (RMSE) of a given regression model versus the number of samples used for training. The RMSE is computed using points evaluated on a grid. For the 2D case, the total number of points is 2601 evenly spaced at an interval of 0.02. The 3D case uses a total of 9261 points with a grid spacing of 0.05. The 4D case uses a total of 14641 points with a spacing of 0.1, and the 5D case uses a total of 7776 validation points with a spacing of 0.2. All plots show the median RMSE of 10 trial runs. We use the median value since, as discussed below, several regression techniques seem to fail for particular sets of samples and the median is more robust against the resulting outliers in the RMSE. For a fixed prediction model, all trials start from the same

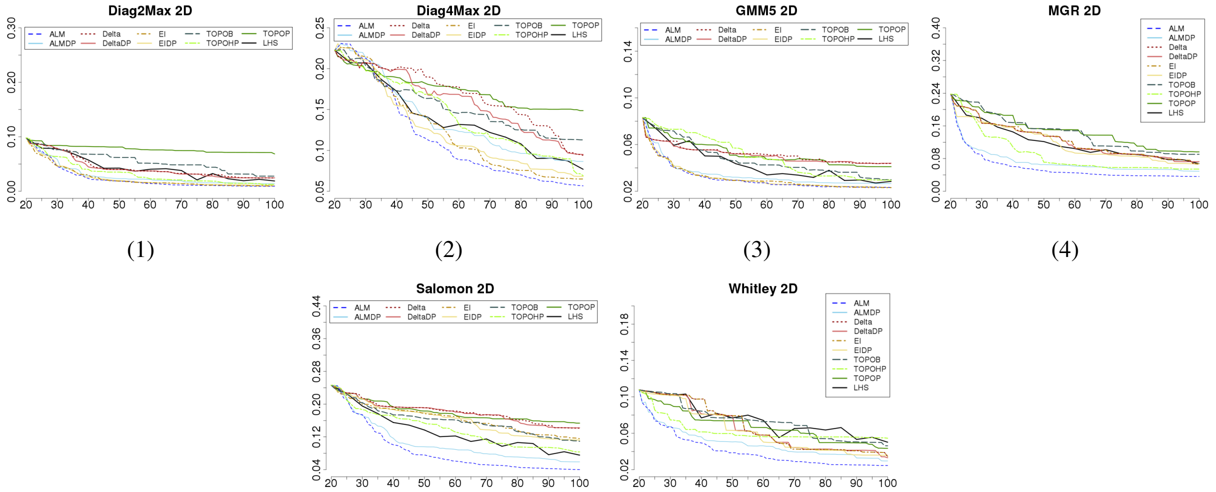
LHS initial training sample $\mathcal{T}$. $|\mathcal{T}| = 20, 30, 100$, and 200 in 2D, 3D, 4D, and 5D respectively. Curves are colored by scoring function with the thicker black line indicating an LHS sample of the given size. During each step of the adaptive sampling, a point is chosen among all $|\mathcal{S}| = 200 * d$ candidate points selected using LHS with the highest score. We have opted to not show variances or percentiles alongside the medians as the resulting plots become too cluttered. Nevertheless, as discussed below even without an explicit representation several of the plots suggest drastic differences in the stability of the regression.



(1) (2) (3) (4)

FIG. 5: Adaptive sampling of the 2D Ackley function using different regression techniques and different scoring functions. (1) GPM; (2) EARTH; (3) MDA; and (4) NNET.

Discussions on Results. Unsurprisingly, the most significant factor in the success of any scoring function is the quality of the regression model. Unfortunately, many experiments even in lower dimensions failed in the sense that the regression based on the space filling design did not converge within the number of samples tried. In fact, for some functions some techniques did not show signs of improvement with an increasing number of samples. While it is possible that these results could be improved through manual parameter tuning, this would likely not be feasible in a real world application where no ground truth is known and the number of samples is typically severely limited. Furthermore, any manual or semi-automatic parameter tuning would make comparing models even more challenging and potentially bias the results. Therefore, we have selected to run all experiments with the default values provide with the various regression packages listed in APPENDIX B. Subsequently, we only consider experiments in which the space filling design indicated a valid surrogate model.

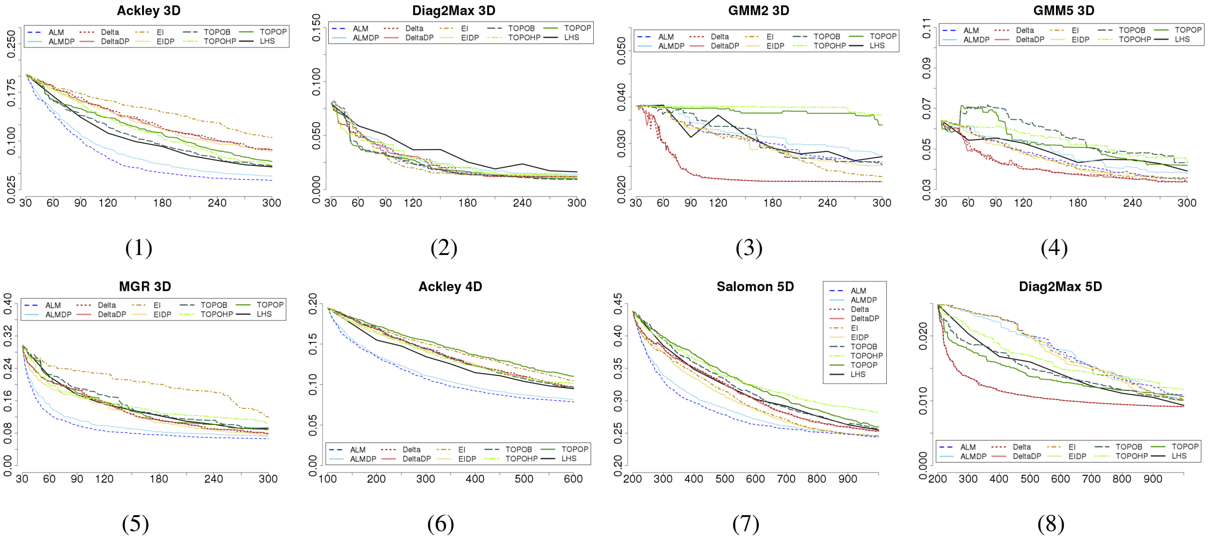


(1) (2) (3) (4)

(5) (6)

FIG. 6: Adaptive sampling various 2D test functions using the GPM. (1) Two maxima Diagonal; (2) Four maxima Diagonal; (3) Mixture of five Gaussians; (4) MGR; (5) Salomon; (6) Whitley.

In general, the GPM based regression produces the smoothest and most distinctive plots showing clear differences between scoring functions. Consider the 2D Ackley function show in Fig. 5(1): The ALM criterion significantly outperforms all other scoring functions with the TopoHP the only other criterion that beats the space filling design. Compare this to the EARTH model shown in Fig. 5(2): Most scoring functions except TopoHP perform qualitatively similar and barely achieve the same quality as the LHS sampling. Furthermore, all curves are less smooth suggesting a high variance among the different trials. The other MARS implementation, MDA, performs slightly worse (Fig. 5(3)) but qualitatively similar with very rough curves without clear trends that fail to achieve the same performance as the space filling sample. Finally, the NNET implementation shown in Fig. 5(4) barely shows any improvement in RMSE with increasing samples, and all sampling strategies including the LHS sample seem to perform similar with large fluctuations. Overall, NNET achieves by far the worst fit at an RMSE almost an order of magnitude larger than that of the GPM model.



(1) (2) (3) (4)

(5) (6) (7) (8)

FIG. 7: Adaptive sampling of various test functions using the GPM. (1) 3D Ackley; (2) 3D two maxima Diagonal; (3) 3D mixture of two Gaussians; (4) 3D mixture of five Gaussians; (5) 3D MGR; (6) 4D Ackley; (7) 5D Salomon; (8) 5D two maxima Diagonal function.

The general differences between regression models seen in the Ackley functions are present for all test functions. Where the GPM models tend to converge for some reasonable number of samples and even in higher dimensions, both MARS implementations have difficulties with non-axis aligned structures such as the Diagonal and the Salomon function. The NNET with the standard parameters consistently performs worse than all other models, and among all techniques only the GPM shows the smoothly converging curves one would traditionally expect. Furthermore, only the GPM model shows significant differences between scoring functions. For the GPM based adaptive sampling the one trend observed in most 2D test functions is that the ALM criterion seems to outperform all others. A possible explanation for this behavior is that the predicted variance on which the ALM criterion is based is an integral part of the GPM model itself. This makes the ALM criterion especially well-suited for a GPM model and could explain the consistently better performance.

Nevertheless, the performance of the scoring functions even for the well behaved GPM regression is far from consistent across test functions. For example, in the Diagonal function with two maxima (Fig. 6(1)) the ALM and EI criteria perform best closely followed by TopoHP. The Delta function is on par with the LHS sampling while the other two topological criteria do not perform well. The four maxima Diagonal version (Fig. 6(2)) shows a similar behavior even though now small differences appear between the ALM and EI criteria and the Delta function now fails to achieve the RMSE of the space-filling sample. The mixture of five Gaussians (Fig. 6(3)) again shows a clear preference for the error driven criteria even though now the Delta function performs worse than the topological scoring functions.

The remaining functions show a very similar pattern with the noticeable exception that now the EI criteria no longer perform as well as the the ALM. For the MGR function (Fig. 6(4)) TopoHP performs second best with EI and Delta mirroring the LHS line. For the Salomon function (Fig. 6(5)) EI performs better than Delta again but both fail to beat the space-filling sample. Finally, the Whitley function (Fig. 6(6)) shows overall a better performance with EI and Delta both outperforming the topological scoring functions.

The 3D experiments using the GPM show largely similar results as shown in Fig. 7. The ALM function generally performs best with the remaining scoring functions in different combinations behind. However, a notable and interesting exception are the mixtures of two and five Gaussians. For both functions the Delta criterion performs significantly better than the rest. One explanation is that these functions are characterized by a few large mountains which, once found, almost entirely determine the function. In such cases the Delta functional may perform well as it picks values purely based on the observed range. It is unclear, however, why such a behavior is not present in the 2D versions.

A similar behavior can be seen in higher dimensions: The 4D Ackley (Fig. 7(6)) and the 5D Salomon function (Fig. 7(7)) show the expected advantage of the ALM criterion while the 5D two maxima Diagonal function (Fig. 7(8)) again prefers the Delta criterion. However, in this case all three topological functions outperform the remaining score functions even though they only perform on par with the LHS samples.

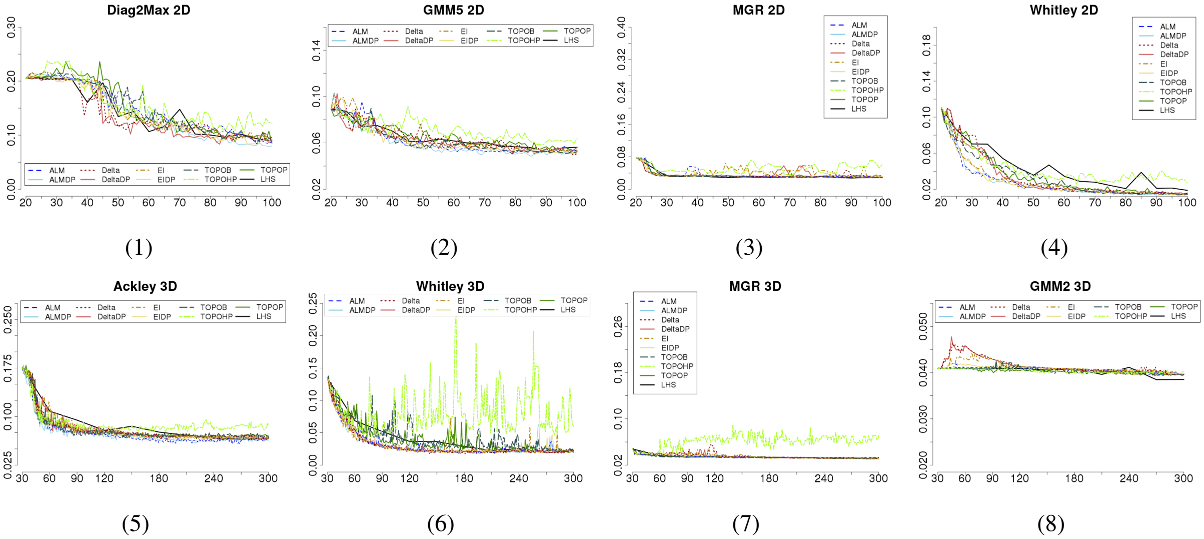
As mentioned above the performance of the other regression techniques is rather spotty and the results of the adaptive scoring experiments are correspondingly inconsistent. The EARTH model produces results similar to the ones shown in Fig. 5(2) in case the model itself converges, for example for the two maxima Diagonal function (Fig. 8(1)) and the mixture of five Gaussians (Fig. 8(2)). However, other experiments show rather large fluctuations in either some adaptive sampling curves, e.g. the TopoHP curve within the MGR function (Fig. 8(3)) or the LHS curve, e.g. the Whitley function (Fig. 8(4)). Overall, the 2D Whitley function shows a good performance with nearly all adaptive criteria outperforming the LHS curve. Interestingly, in its 3D incarnation (Fig. 8(6)) the adaptive sampling does not nearly work as well and furthermore shows excessive variations in several curves.

In general there appears to be no significant advantage of one scoring function over another even though for the EARTH model in many cases the TopoHP seems to perform especially bad, for example, for the 3D Whitley and 3D MGR function (Fig. 8(6) and Fig. 8(7)). This is rather surprising since for the GPM model TopoHP performed rather well and typically better than the other topological scoring functions. The fact that there exist virtually no difference among the other scoring functions is likely due to the error computation. For non-GPM models like MARS the expected prediction error that forms the bases for both the ALM and the EI criterion is constructed through bootstrapping and thus is probably less reliable. This may negate the differences between error based criteria and the others, and result in across-the-board worse performance. Unfortunately, we have not been able to get acceptable results for dimensions beyond three as even for simple models like the mixture of two Gaussians the non-GPM models did not converge. A failed but nevertheless interesting result comes from the mixture of two Gaussians (Fig. 8(8)) where the model in general shows no real improvements for higher number of points but both the Delta and the EI criteria first decrease the quality of the fit.

The other MARS implementation MDA, unsurprisingly, shows largely the same behavior as EARTH. Fig. 9 shows four examples each in two and three dimensions. Comparing with the Earth model, we could see typically comparable RMSE, and higher variability (e.g. Fig. 9(1) and Fig. 9(2)) with the exception of 3D Whitley function (Fig. 9(6)). Again the 3D mixture of two Gaussians (Fig. 9(8)) shows the initial decrease in RMSE for several scoring functions.

Finally, the NNET implementation in its default setting performs clearly worse with excessive variability in all plots which makes interpretation of any possible trends questionable. Apart from the Ackley function of Fig. 5(4) some models that performed reasonably in two dimensions are the two maxima Diagonal function (Fig. 10(1)), mixture of five Gaussians (Fig. 10(2)), and the Salomon function (Fig. 10(4)). The MGR function (Fig. 10(3)) shows a competitive RMSE as baseline but excessive variations for virtually all adaptive scoring functions as well as the LHS line. The 3D MGR function (Fig. 11(2)) shows a similar behavior except that the LHS line now appears to be stable.

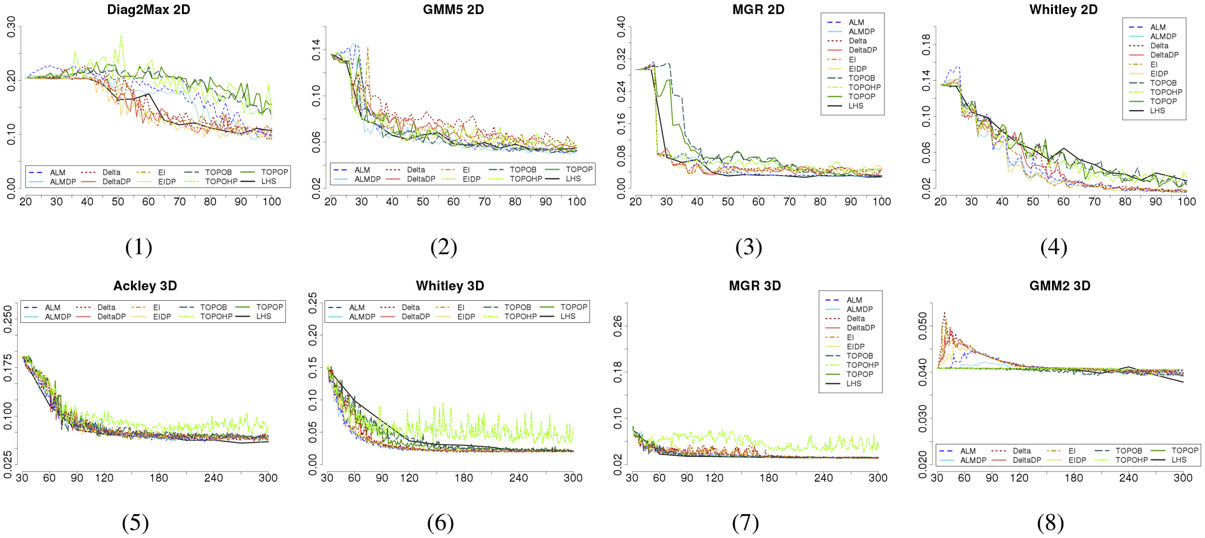
The 3D Ackley (Fig 11(1)) and Salomon function (Fig. 11(3)) both show the high variability and high RMSE with no adaptive scoring functions able to outperform the space-filling sampling. Just as with the two MARS implementations the 3D mixture of two Gaussians (Fig. 11(4)) does not improve with increasing number of samples and shows the initial increase in RMSE for some scoring functions.



(1) (2) (3) (4)

(5) (6) (7) (8)

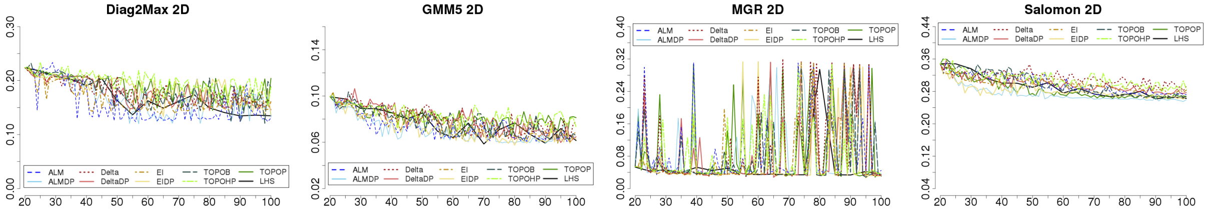
FIG. 8: Adaptive sampling of various test functions using the EARTH regression model. (1) 2D two maxima Diagonal function; (2)2D mixture of five Gaussians; (3)2D MGR function; (4) 2D Whitley functions; (5) 3D Ackley function; (6) 3D Whitley function; (7) 3D MGR function; (8) 3D mixture of two Gaussians.



(1) (2) (3) (4)

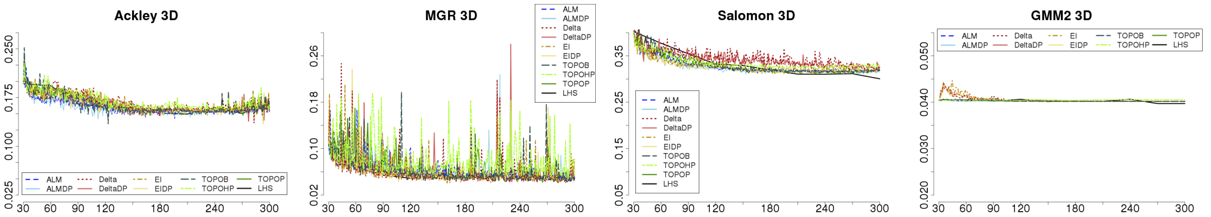
(5) (6) (7) (8)

FIG. 9: Adaptive sampling of various test functions using the MDA model. (1) 2D two maxima Diagonal; (2) 2D mixture of five Gaussians; (3) 2D MGR function; (4) 2D Whitley function; (5) 3D Ackley function; (6) 3D Whitley function; (7) 3D MGR function; (8) 3D mixture of two Gaussians.



(1) (2) (3) (4)

FIG. 10: Adaptive sampling of various 2D test functions using the NNET model. (1) Two maxima Diagonal function; (2) Mixture of five Gaussians; (3) MGR function; (4) Salomon function.



(1) (2) (3) (4)

FIG. 11: Adaptive sampling of various 3D test functions using the NNET model. (1) Ackley function; (2) MGR function; (3) Salomon function; (4) Mixture of two Gaussians.

5. DISCUSSION

After running an extensive set of experiments in two to five dimensions with nine different scoring functions some general trends appear even though the fundamental question of which particular scoring function to use remains largely inconclusive. The results given in the paper is only a small first step in the this direction. The first, not necessarily surprisingly result is that the quality of the underlying regression model plays a key role in the performance of any adaptive sampling technique. In this study the GPM model performs the best and is the only one showing significant differences between scoring functions. Overall, it seems that the combination of ALM and GPM is the preferred choice even though some functions perform well with the Delta criteria.

The remaining regression models all show problems fitting many of the test functions and the large variability suggest that they are sensitive to specific sample locations. In the cases where reasonable fits have been achieved all scoring functions perform equally well (or poorly).

The new topological scoring functions are largely competitive in terms of RMSE and often perform among the top scoring functions. Some results suggest that detailed fits with a high number of samples are less well suited for topological scoring functions as they are designed to recover larger scale features. Nevertheless, topological scoring functions are expected to better recover the global structure of a function, and finding quantitative metrics to test this hypothesis as well as expanding the class of such functions will be the focus of future research.

ACKNOWLEDGMENTS

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-JRNL-519591. In particular the *Uncertainty Quantification Strategic Initiative Laboratory Directed Research and Development Project* at LLNL under project tracking code 10-SI-013.

REFERENCES

1. McKay, M. D., Beckman, R., and Conover, W. J., A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics*, 21:239–245, 1979.
2. Tang, B., Orthogonal array-based latin hypercubes, *Journal of the American Statistical Association*, 88(424):1392–1397, 1993.
3. Dam, E. R.v., Two-dimensional minimax latin hypercube designs, *Discrete Applied Mathematics*, 156:3483–3493, 2008.
4. Rasmussen, C. E. and Williams, C. K. I., *Gaussian Processes for Machine Learning*, MIT Press, 2006.
5. Friedman, J., Multivariate adaptive regression splines, *The Annals of Statistics*, 19(1):1–67, 1991.
6. Ripley, B. D., *Pattern Recognition and Neural Networks*, Cambridge University Press, 1996.
7. Fang, K.-T., Li, R., and Sudjianto, A., *Design and Modeling of Computer Experiments*, Chapman and Hall/CRC, 2005.
8. Sacks, J., Schiller, S. B., and Welch, W., Design for computer experiments, *Technometrics*, 31:41–47, 1989.
9. Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning*, Springer-Verlag, 2009.
10. Davison, A. C. and Hinkley, D. V., *Bootstrap Methods and their Application*, Cambridge University Press, 1997.
11. Santner, T. J., Williams, B., and Notz, W. I., *The Design and Analysis of Computer Experiments*, Springer-Verlag, 2003.
12. Koehler, J. R. and Owen, A. B. Computer experiments. In: Ghosh, S. and Rao, C. R. (Eds.), *Handbook of Statistics*. Elsevier Science, 1996.
13. Bates, R. A., Buck, R. J., Riccomagno, E., and Wynn, H. P., Experimental design and observation for large systems, *Journal of the Royal Statistical Society, Series B: Methodological*, 58:77–94, 1996.
14. Schonlau, M. Computer Experiments and Global Optimization. PhD thesis, University of Woterloo, 1997.
15. Jones, D., Schonlau, M., and Welch, W., Efficient global optimization of expensive black-box functions, *Journal of Global Optimization*, 13:455–492, 1998.
16. Lam, C. Q. Sequential Adaptive Designs In Computer Experiments For Response Surface Model Fit. http://etd.ohiolink.edu/, 2008.
17. Carr, H., Snoeyink, J., and Axen, U., Computing contour trees in all dimensions, *Computational Geometry Thoery and Applications*, 24(3):75–94, 2003.
18. Reeb, G., Sur les points singuliers d'une forme de pfaff complément intégrable ou d'une fonction numérique, *Comptes Rendus de L'Académie ses Séances Paris*, 222:847–849, 1946.
19. Pascucci, V., Scorzelli, G., Bremer, P.-T., and Mascarenhas, A., Robust on-line computation of reeb graphs: simplicity and speed, *ACM Transactions on Graphics*, 26(3):58, 2007.
20. Boyell, R. L. and Ruston, H., Hybrid techniques for real-time radar simulation, In *Proceedings Fall Joint Computer Conference*, pp. 445–458, 1963.
21. Edelsbrunner, H., Harer, J., and Zomorodian, A. J., Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds, *Discrete and Computational Geometry*, 30:87–107, 2003.
22. Gyulassy, A., Natarajan, V., Pascucci, V., and Hamann, B., Efficient computation of Morse-Smale complexes for three-dimensional scalar functions, *IEEE Transactions on Visualization and Computer Graphics*, 13:1440–1447, 2007.
23. Carr, H., Snoeyink, J., and Panne, M.v. d., Simplifying flexible isosurfaces using local geometric measures, In *Proceedings 15th IEEE Visualization*, pp. 497–504, 2004.
24. Laney, D., Bremer, P.-T., Mascarenhas, A., Miller, P., and Pascucci, V., Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities, *IEEE Transactions on Visualization and Computer Graphics*,

12:1052–1060, 2006.

25. Bremer, P.-T., Weber, G., Pascucci, V., Day, M., and Bell, J., Analyzing and tracking burning structures in lean premixed hydrogen flames, *IEEE Transactions on Visualization and Computer Graphics*, 16(2):248–260, 2010.
26. Gyulassy, A., Duchaineau, M., Natarajan, V., Pascucci, V., E.Bringa, , Higginbotham, A., and Hamann, B., Topologically clean distance fields, *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1432–1439, 2007.
27. Oesterling, P., Heine, C., Jaenicke, H., Scheuermann, G., and Heyer, G., Visualization of high-dimensional point clouds using their density distribution's topology, *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1547–1559, 2011.
28. Correa, C. D. and Lindstrom, P., Towards robust topology of sparsely sampled data, *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2011.
29. Edelsbrunner, H., Harer, J., Natarajan, V., and Pascucci, V., Morse-Smale complexes for piecewise linear 3-manifolds, In *Proceedings 19th Annual Symposium on Computational Geometry*, pp. 361–370, 2003.
30. Chazal, F., Guibas, L. J., Oudot, S. Y., and Skraba, P., Analysis of scalar fields over point cloud data, In *Proceedings 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1021–1030, 2009.
31. Gerber, S., Bremer, P.-T., Pascucci, V., and Whitaker, R. T., Visual exploration of high dimensional scalar functions, *IEEE Transactions on Visualization and Computer Graphics*, 16:1271–1280, 2010.
32. Gerber, S., Rübel, O., Bremer, P.-T., Pascucci, V., and Whitaker, R. T., Morse-Smale regression, *Journal of Computational and Graphical Statistics*, 2012, in press.
33. Vedaldi, A. and Soatto, S., Quick shift and kernel methods for mode seeking, In *Proceedings European Conference on Computer Vision*, pp. 705–718, 2008.
34. Edelsbrunner, H., Letscher, D., and Zomorodian, A. J., Topological persistence and simplification, *Discrete and Computational Geometry*, 28:511–533, 2002.
35. Carlsson, G., Zomorodian, A. J., Collins, A., and Guibas, L. J., Persistence barcodes for shapes, In *Proceedings Eurographs/ACM SIGGRAPH Symposium on Geometry Processing*, pp. 124–135, 2004.
36. Silva, V.d. and Ghrist, R., Coverage in sensor networks via persistent homology, *Algebraic and Geometric Topology*, 7:339–358, 2007.
37. Carlsson, E., Carlsson, G., and Silva, V.d., An algebraic topological method for feature identification, *International Journal of Computational Geometry and Applications*, 16:291–314, 2003.
38. Edelsbrunner, H. and Harer, J., Persistent homology - a survey, *Contemporary Mathematics*, 453:257–282, 2008.
39. Carlsson, G., Ishkhanov, T., Silva, V.d., and Zomorodian, A., On the local behavior of spaces of natural images, *International Journal of Computer Vision*, 76:1–12, 2008.
40. Kloke, J. and Carlsson, G. Topological de-noising: strengthening the topological signal. Manuscript, 2010.
41. Bremer, P.-T., Edelsbrunner, H., Hamann, B., and Pascucci, V., A topological hierarchy for functions on triangulated surfaces, *IEEE Transactions on Visualization and Computer Graphics*, 10(385-396), 2004.
42. Gyulassy, A., Natarajan, V., Pascucci, V., Bremer, P. T., and Hamann, B., Topology-based simplification for feature extraction from 3D scalar fields, In *Proceedings 16th IEEE Visualization*, pp. 535–542, 2005.
43. Cole-McLaughlin, K., Edelsbrunner, H., Harer, J., Natarajan, V., and Pascucci, V., Loops in reeb graphs of 2-manifolds, In *Proceedings 19th Annual Symposium on Computational Geometry*, pp. 344–350, 2003.
44. Chazal, F., Cohen-Steiner, D., Glisse, M., Guibas, L. J., and Oudot, S. Y., Proximity of persistence modules and their diagrams, In *Proceedings 25th Annual Symposium on Computational Geometry*, pp. 237–246, 2009.
45. Cohen-Steiner, D., Edelsbrunner, H., and Harer, J., Stability of persistence diagrams, *Discrete and Computational Geometry*, 37:103–120, 2007.
46. MacKay, D., Information-based objective functions for active data selection, *Neural Computation*, 4(4):589–603,

1992.

47. Ackley, D. H., *A connectionist machine for genetic hillclimbing*, Kluwer Academic Publishers, 1987.

48. Grollman, D. Sparse online gaussian process. http://lasa.epfl.ch/ dang/code.shtml, 2011.

49. Csató, L. and Opper, M., Sparse online gaussian processes, *Neural Computation*, 14:641–668, 2002.

50. Csató, L., Gaussian processes - iterative sparse approximations, PhD thesis, Aston University, 2002.

51. Milborrow, S. *earth: Multivariate Adaptive Regression Spline Model*, 2011. R package version 3.2-1, Derived from mda:mars by T. Hastie and R. Tibshirani.

52. Hastie, T., Tibshirani, R., Leisch, F., Hornik, K., and Ripley, B. D. *mda: Mixture and flexible discriminant analysis*, 2011. R package version 0.4-2.

53. Venables, W. N. and Ripley, B. D., *Modern Applied Statistics with S*, Spinger, 2002.

54. Carnell, R. *lhs: Latin Hypercube Samples*, 2009. R package version 0.5.

APPENDIX A. EXAMPLE TEST FUNCTIONS CLOSED FORMS AND PLOTS

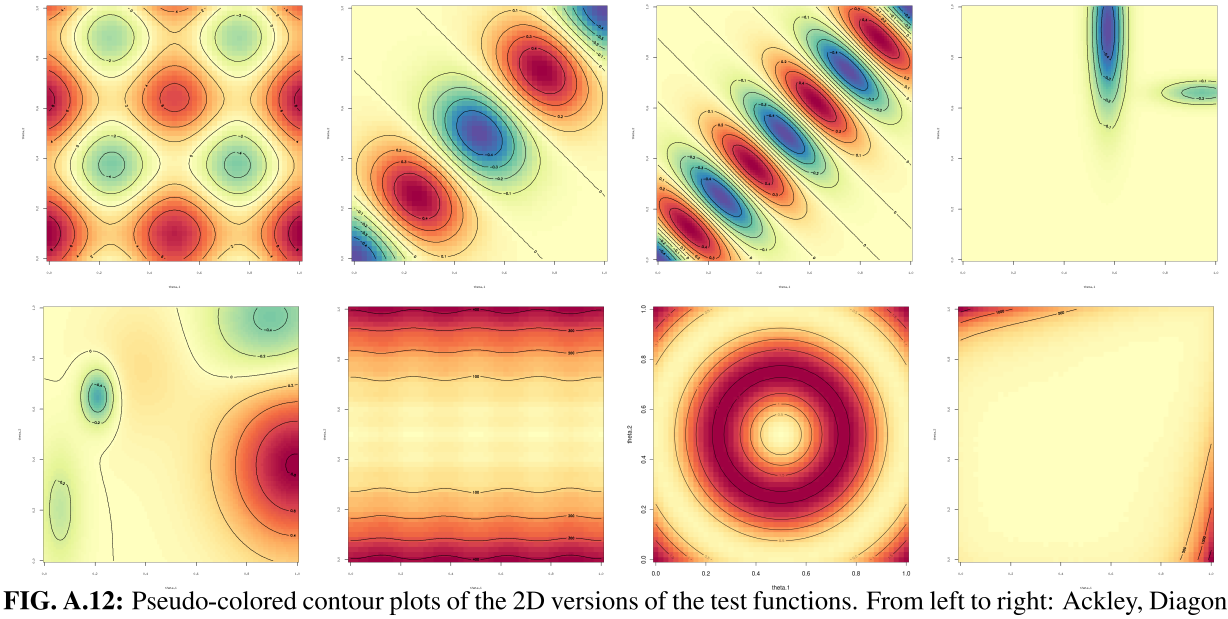


FIG. A.12: Pseudo-colored contour plots of the 2D versions of the test functions. From left to right: Ackley, Diagonal function with 2 maxima (Diag2Max), Diagonal function with 4 maxima (Diag4Max), Gaussian Mixture Model with 2 (GMM2) and 5 Gaussians (GMM5), Mis-scaled Generalized Rastrigin (MGR), Salomon and Whitley, respectively.

We use several testing functions with closed forms described below. All functions can be generalized to higher dimensions and their two dimensional contour plots are shown in Fig A.12. Let D be the dimension of the input vector $\vec{\theta}$.

Ackley.

$$f(\vec{\theta}) = \sum_{i=1}^{D-1} \left(e^{-0.2} \sqrt{\theta_i^2 + \theta_{i+1}^2} + 3(\cos(2\theta_i) + \sin(2\theta_{i+1})) \right),$$

where $\{\vec{\theta} | \theta_i \in [-3, 3]\}$.

Diagonal. Let m be the number of maxima along the main diagonal,

$$f(\vec{\theta}) = \tfrac{1}{2}\sin\left(\pi\left(\tfrac{1}{2} + ((m+1) \bmod 2) + \tfrac{m(\sum_{i=1}^{D}\theta_i)}{D}\right)\right) \times e^{\left(\left(\sum_{i=1}^{D}\theta_i^2 - \frac{\left(\sum_{i=1}^{D}\theta_i\right)^2}{D}\right)\left(\frac{\log(0.001)}{\sqrt{D}}\right)\right)},$$

where $\{\vec{\theta}|\theta_i \in [-1,1]\}$.

Random Gaussian Mixture Model.

Let m be the number of extrema in the domain. Let a_i be the amplitude of the i^{th} extrema. Let $c_{i,j}$ be the j^{th} coordinate of the i^{th} extrema. Let $\sigma_{i,j}$ be the standard deviation of the i^{th} extrema with respect to the j^{th} coordinate.

$$f(\vec{\theta}) = \sum_{i=1}^{m}\left(a_i e^{-\left(\sum_{j=1}^{D}\frac{(\theta_{i,j}-c_{i,j})^2}{2\sigma_{i,j}^2}\right)}\right)$$

where $\{\vec{\theta}|\theta_i \in [0,1]\}$.

Mis-scaled Generalized Rastrigin.

$$f(\vec{\theta}) = 10D + \sum_{i=1}^{D}\left((10^{\frac{i-1}{D-1}}\theta_i)^2 - 10\cos(2\pi(10^{\frac{i-1}{D-1}}\theta_i))\right)$$

where $\{\vec{\theta}|\theta_i \in [-2,2]\}$.

Salomon.

$$f(\vec{\theta}) = -\cos\left(2\pi\sum_{i=1}^{D}\theta_i^2\right) + 0.1\sqrt{\sum_{i=1}^{D}\theta_i^2} + 1$$

where $\{\vec{\theta}|\theta_i \in [-1,1]\}$.

Whitley.

$$f(\vec{\theta}) = \sum_{i=1}^{D}\sum_{j=1}^{D}\left(\frac{k_{ij}^2}{4000} - cos(k_{ij}) + 1\right)$$

where $\{\vec{\theta}|\theta_i \in [-1,2]\}$, and $k_{ij} = (100(\theta_i^2 - \theta_j)^2 + (1-\theta_j)^2)$.

APPENDIX B. SOFTWARE PACKAGES AND PARAMETER SETTINGS

Now we give details on packages and parameter settings used in our experiments.

For GPM, we use the `Sparse Online Gaussian Process` (SOGP) C++ library [48], which is based on work in [49, 50]. The default parameters are employed, see [50] for details, i.e. we use the Radial Basis Kernel with a spherical covariance set to $\sigma_0^2 = 0.1$, the widths are uniformly set to 0.1, and the amplitude $A = 1$.

For EARTH, MDA and NNET, we use non-parametric bootstrapping using 250 samples without cross-validation.

For EARTH, we use the `earth` library in `R` [51]. We use the following parameter settings, see [51] for details. If a parameter is not listed, the default value given by the package is used.

```
degree=3 // Maximum degree of interaction (Friedman's mi)
nk=63  //Maximum number of model terms before pruning
minspan=1 //Min. dist. between knots, 1 for non-noisy data
thresh=1e-8  //Forward stepping threshold
penalty=3 //Generalized Cross Validation Penalty per knot
```

For MDA, we use `mda` library in `R`, with the following parameters (see [52] for details, non-listed parameters use package defaults),

```
degree=3 // Maximum degree of interaction (Friedman's mi)
nk=63  //Maximum number of model terms
thresh=1e-8 // Forward stepping threshold
penalty=3 // The cost per degree of freedom charge
```

For NNET, we use `nnet` library in `R`, with the following parameters (see [53] for details, non-listed parameters use package defaults),

```
size=1+ceiling(sqrt(D)) // Number of units in the hidden layer
                        // D is the dimensionality of the input
decay=1e-3 // Parameter for weight decay
skip=TRUE // Add skip-layer connections from input to output
linout=TRUE // Linear output units, as opposed to logistic
```

To create the space filling samplings, the `lhs` library in `R` [54] is used. More specifically, we use the `randomLHS` function, which chooses uniform, random samples without any attempts to optimize the design, to construct the training data, candidate data, and LHS samples in the plots shown.