SCI INSTITUTE
TECHNICAL REPORT

Topology-Based Active Learning

Dan Maljovec, Bei Wang, John Moeller, Valerio Pascucci

UUSCI-2014-001

Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112 USA

July 28, 2014

Abstract:

A common problem in simulation and experimental research involves obtaining time-consuming, expensive, or potentially hazardous samples from an arbitrary dimension parameter space. For example, many simulations modeled on supercomputers can take days or weeks to complete, so it is imperative to select samples in the most informative and interesting areas of the parameter space. In such environments, maximizing the potential gain of information is achieved through active learning (adaptive sampling). Though the topic of active learning is well-studied, this paper provides a new perspective on the problem. We consider topologybased batch selection strategies for active learning which are ideal for environments where parallel or concurrent experiments are able to be run, yet each has a heavy cost. These strategies utilize concepts derived from computational topology to choose a collection of locally distinct, optimal samples before updating the surrogate model. We demonstrate through experiments using a several different batch sizes that topology-based strategies have comparable and sometimes superior performance, compared to conventional approaches.

THE UNIVERSITY OF UTAH

Topology-Based Active Learning

Dan Maljovec
School of Computing
University of Utah
Salt Lake City, UT 84112
maljovec@cs.utah.edu

Bei Wang
Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112
beiwang@sci.utah.edu

John Moeller
School of Computing
University of Utah
Salt Lake City, UT 84112
moeller@cs.utah.edu

Valerio Pascucci
Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112
pascucci@sci.utah.edu

Abstract

A common problem in simulation and experimental research involves obtaining time-consuming, expensive, or potentially hazardous samples from an arbitrary dimension parameter space. For example, many simulations modeled on supercomputers can take days or weeks to complete, so it is imperative to select samples in the most informative and interesting areas of the parameter space. In such environments, maximizing the potential gain of information is achieved through active learning (adaptive sampling). Though the topic of active learning is well-studied, this paper provides a new perspective on the problem. We consider topology-based batch selection strategies for active learning which are ideal for environments where parallel or concurrent experiments are able to be run, yet each has a heavy cost. These strategies utilize concepts derived from computational topology to choose a collection of locally distinct, optimal samples before updating the surrogate model. We demonstrate through experiments using a several different batch sizes that topology-based strategies have comparable and sometimes superior performance, compared to conventional approaches.

1 Introduction

Active learning (or adaptive sampling) has been utilized in many settings to solve a variety of problems as diverse as: nuclear safety simulations [36], network queueing [40], environmental monitoring [24], and parameter setting in cosmological models [8]. Because samples can be expensive, time-consuming, hazardous, or otherwise difficult to obtain, ideally one would wish to gain the most information about the phenomenon under study with the least amount of sampled data. By utilizing all of the information we currently know about a phenomenon, we drive the investigation toward the most interesting or least understood areas of the simulation space.

Many such experiments exhibiting these characteristics have the advantage that samples can be acquired simultaneously through the use of multiple data gathering resources (parallel computing environment, existence of multiple sensory equipment,etc.). Thus, utilizing an active learning framework that can select multiple informative points would yield a greater return on investment, than either relying on space-filling designs or selecting a single point and waiting for a response until the next single point is selected. Efforts have been made to perform such batch selection before, but we present a novel perspective of studying the topology of the scoring function during such a selection process. Our method does not try to redesign the scoring function, but instead relies on the *topology*

of the scoring function to generate samples from informative, but distinct areas of interest. In this way, our method is agnostic of the surrogate model and scoring function being used. We validate our method on several test datasets where we look at two different active learning problems. The first is an attempt to create a globally adequate fit of our surrogate model by reducing the amount of error over the entire input parameter space, and the second problem is that of attempting to locate an isocontour of our domain known as the *limit surface*.

2 Related work

2.1 Active Learning

Active learning (see [44] for a survey) has been used in many contexts and has various other names such as adaptive sampling, sequential sampling, and optimal experimental design. The basic process of active learning tries to converge to an acceptable fitting of the surrogate model using fewer data points than using a standard space-filling design. Techniques for achieving this include reducing global uncertainty [49, 9], exploiting areas of high gradient [35], and exploring areas where information gain is optimal [32]. A slightly modified version is aimed at global optimization rather than global accuracy of the model [38, 31]. The active search method takes a slightly different approach by looking at the problem as attempting to sample points contained in a single class [21]. This last case is related to our second test problem of limit surface recovery where we are interested in defining the boundary between two classes.

2.2 Limit Surface Recovery

The limit surface recovery problem in active learning is: can one identify and characterize the levelset defined by a given threshold value of the simulation output. In reliability engineering, this problem is known as determining the limit state function and can be thought of as the point where a system's load exceeds its capacity where load and capacity are functions of the input parameters. The literature in this field attempts to either converge to a single most probable point (MPP) through nonlinear optimization methods [20, 19] or through adaptive sampling [50, 13, 52, 4].

In the statistics literature, [40] has attempted to solve this problem by defining an expected improvement candidate scoring function that harnesses both a prediction's proximity to the threshold value and also the uncertainty associated with the prediction. In similar spirit, the machine learning community has designed similar yet simpler scoring functions that take advantage of similar information. [8] defines a scoring function called the straddle function, and [24] introduced the ambiguity scoring function which can be thought of as a generalized version of the straddle function. All three methods utilize a Gaussian process surrogate model, however the latter two are performed in what [44] called the pool-based active learning setting, whereas the first uses a global optimization to select the optimal point at each iteration.

2.3 Surrogate Modeling

Neural networks. Regression modeling using neural networks goes back at least 20 years, with notable examples being [46] and [43], and of course the book by [5].

Support vector machines. The use of *support vector machines* (SVMs) for regression is almost as venerable. In theory, support vector classification and support vector regression differ only in how the learned function is used. Whereas a classifier thresholds the value of the function, in regression we would want to keep the entire function. The measure of loss is typically least squares, whereas in classification different loss functions are used. See [45, 48, 47, 14] to name a few.

Regression SVMs are an obvious choice in level detection in the surrogate modeling problem, because they approximate the function under query. Classification SVMs can also be utilized, however, by labeling sampled points according to whether they exhibit values above or below the threshold value. Then, after the model is trained, the level set is assumed to be the decision boundary of the model.

Bayesian methods. Not surprisingly, there are a set of Bayesian techniques that tackle the same set of problems, but focus on *inference*. A prior distribution is also employed before sampling to express

the belief about the shape of the data. There is a rich body of work that discusses these techniques, but for an overall discussion, see [33]. For a discussion of using Gaussian process (GP) models in a Bayesian regression framework, see [41].

2.4 Batch Sampling

We define batch sampling as the process of selecting b points, where $b > 1$ in an active learning framework before querying the oracle and updating the surrogate model. [44] provides a summary of different attempts to draw multiple candidates at a time. SVM-based approaches are explored in [6, 51]. [23], [1], [3], and [12] use a GP in a batch sampling setting where the goal is to maximize the GP rather than gain a global understanding. [23], [3], [12], and [24] all take advantage of the fact that the variance estimate of the GP only relies on the location of training values and not the true response values whereby they can greedily choose a batch set of points by updating their GP after each point is selected and rescoring the remaining candidates. [28, 29] exploit a different measure of uncertainty by minimizing the Fisher information matrix to achieve high quality batch samples. [26] and [25] look at the selection process as an optimization and settle on local solutions as the full optimization in each case is NP-hard. [2] extends the work of [1].

Many of the techniques described here are based on binary logistic regression, and are used to solve classification problems, where this is similar to our problem of identifying the limit surface, we are also interested in determining a global best fit. As we are utilizing a GP, we compare our methods to that used by [23] and [24].

3 Technical Background

Test Problems. We consider two test problems on which we evaluate our methods. Both are formulated as pool-based sampling (e.g. [44]) problems and involve the same pipeline of scoring a candidate set of points and selecting a batch of size b locations to query our true function which are then added to our training set, and the process is repeated on the remaining candidates. The difference of the two methods lies in how we score the points. In the first problem considered, the global recovery problem where the idea is to obtain an accurate surrogate over the entire input domain, we have chosen to use the active learning McKay (ALM) algorithm which directly uses the predicted variance from our GP as the score for any given location.

In the problem of limit surface recovery, where we would like to recover the level set of a scalar function $f(\vec{x})$ at a given threshold t (i.e. the limit surface), we use a different strategy. Here, we are less concerned with convergence of the global response surface, but rather we are interested in characterizing a single implicit limit surface. The *straddle* scoring function [8, 7] (and its variation in [24]) is specially designed to target the limit surface recovery problem, which scores points based on a mixture of high predicted variance and a predicted mean near the threshold value. The exact formula is: $\textbf{straddle}(\vec{x}) = 1.96\ \hat{\sigma}(\vec{x}) - |\hat{\mu}(\vec{x}) - t|$, where $\vec{x}$ is the input vector, $\hat{\mu}$ and $\hat{\sigma}$ represent the predicted mean and variance of the Gaussian process *at* $\vec{x}$, and t represents the threshold value of interest. We use the straddle scoring function to study the limit surface recovery problem. As we can see, both scoring functions are based on the predicted variance of the GP which only relies on the location of true responses and not on the true response. This allows us a fair comparison to the technique used in [23], [3], [12], and [24]. However, we will note that our pure topology-based method is agnostic of the scoring function and therefore more robust than these methods since it can be used in settings where targeting the high-variance may not be ideal.

Morse-Smale Complex and its Approximations. Topological structures such as Reeb graphs [42, 39] and Morse-Smale complexes (MSCs)[17, 16, 27] have been used to describe the topology of scalar functions defined on point cloud data. The main idea behind the Morse-Smale complex of a function f is that it partitions the domain of f into regions of uniform gradient behavior. Formally, let f be a smooth function on $\mathbb{R}^n$. A point in the domain is *crtitial* if its gradient is zero, otherwise it is *regular*. At a regular point, the gradient is well-defined and integrating it traces out an integral line. The function increases along the integral line, which begins at a local minimum and ends at a local maximum of f. The ascending/descending manifold of a critical point x is defined as all points whose integral lines start/end at x. The set of intersections of ascending and descending manifolds creates the Morse-Smale complex of f, where each *cell* of the complex is a union of integral lines

that all share the same origin and destination. Figure 1 illustrates a MSC for a 2D function and the corresponding gradient behaviors.

The MSC is useful in this context as it helps us understand the topology of a scoring function, which is a scalar function, under the active learning setting. Computing the MSC associated with a scoring function identifies the local maxima of the function, that is, the points which have zero gradient and whose corresponding function values are higher than their neighbors. Another important property associated with MSC is that we could associate an importance measure, referred to as the *persistence*, to each critical points of f. The persistence of a critical point, in a nutshell, indicates the importance of its associated topological features. We therefore could use the notion of persistence simplification [18], whereby less salient (important) features are merged with neighboring, more significant features. Such a procedure is illustrated in 1(d)-(e).

We use a particular variation of MSC developed in [22] that approximate MSC in high-dimensions based on point cloud data. The main idea is to impose a neighborhood graph (e.g. relaxed Gabriel graph [10]) on the point cloud and represent the gradient at each point by its adjacent, steepest ascending edge, based on function values of its neighboring points in the graph. Such an approximation has been used previously in active learning setting, through the design of scoring functions [35]. An initial study of limit surface recovery based on MSC has appeared in [36].

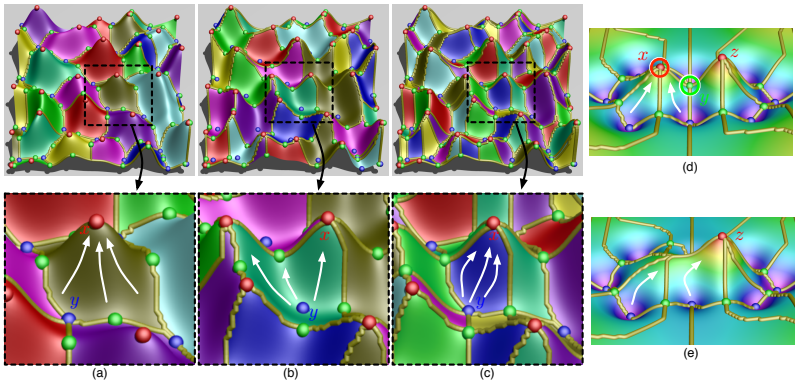

Figure 1: (Figure reproduced from [37].) For a height function defined on a 2D domain (where maxima, minima and saddles are colored red, blue and green respectively): (a) Descending manifold: gradient flow (white arrow) ends at the same maximum; (b) Ascending manifold: gradient flow starts at the same minimum; (c) For each Morse-Smale cell, the gradient flow begins and ends at the same maximum-minimum pair. **Persistence simplification**: In (d), the left peak at the maximum x is less important topologically than its nearby peak at maximum z, since x is lower. We convert this feature to a single peak, as shown in (e), by redirecting gradient flow (white arrow) from x to z. This simplifies the function by canceling the local maximum x with its nearby saddle y.

4 Method

4.1 Batch Selection for Active Learning: General Pipeline

The general pipeline of batch selection begins with a given set of training data with true response values. First, a response surface surrogate model is fit to the training data. Second, a set of candidate points is chosen in the domain based on a certain sampling technique, and the surrogate model is evaluated at these locations, obtaining a set of approximated values. Third, each candidate point is assigned a score based on some scoring function. Finally, a number of candidates (i.e. up to the batch size) with the highest scores are selected, evaluated (to obtain true response values), and added to the training data to begin a new around of fitting. This pipeline is given in Algorithm 1, where we replace **SELECT** with various selection strategies. We use the sparse online Gaussian process [11] as the surrogate model. For a given candidate, such a Gaussian process model gives a (mean) prediction value and a variance associated with the prediction. The set of candidate points and the initial training data are selected using a Centroidal Voronoi Tessellations (CVT) sampling strategy [15]. A CVT sampling strategy gives high-quality samples without the biases introduced by a uniform grid sampling, although it is cost prohibitive in high dimensions. A Latin hypercube

sampling (LHS) [30] strategy can be used for high-dimensional examples. For each dataset, we measure the accuracy of our global or local fitting by comparing to the ground truth at each iteration, under different metrics (e.g. root-mean-square error, Hausdorff distance, F1-score).

For the global surface recovery problem, we utilize the *active learning McKay* (ALM) scoring function [34] that uses the predicted variance from the surrogate model as the score for a candidate point. In this way, we reduce the overall uncertainty of the surrogate model by placing samples in the areas of the highest variance.

For the limit surface recovery problem, we utilize the *straddle* scoring function [8]. For a given candidate point, its score is formulated as a linear combination of the distance between its predicted value and the limit surface threshold value, and the variance. In other words, a candidate will have a high score if its predicted value is near the limit surface threshold and/or it exhibits a large prediction variance.

Algorithm 1 Batch selection for active learning

INPUT: a set of training points, T
a set of candidate points, C
real-valued scoring function, $S(M, x)$, where M is a surrogate model, x is a point
real-valued true response function, $f(x)$, where x is a point
batch size, b
OUTPUT: A surrogate model, $M(x)$, that evaluates at any point x

while $C \neq \emptyset$ and not converged **do**
  $M \leftarrow [T, f(T)]$ ▷ Build a surrogate M with the training data
  $R \leftarrow$ **SELECT**(C, S, M, b) ▷ This is the step we will be augmenting
  $C \leftarrow C \setminus R$ ▷ Remove selected candidates
  $T \leftarrow T \cup R$ ▷ Add candidates to the training data
end while

4.2 Batch Selection Strategies

Naive Batch Selection. Batch selection is simple and straightforward when the batch size $b = 1$, where the point with the highest score is selected. When $b > 1$, it is not clear whether selecting the top b highest scoring candidates is desirable. For example, if the scoring function is based on predicted variance (e.g. Active Learning McKay, a.k.a., ALM), the selected candidates may be clustered in a single area of high variance. However, evaluating the function value at a single location in that area would suffice to reduce the uncertainty for the whole area. Nonetheless, we use this naive strategy as a baseline and its corresponding algorithm is given in Algorithm 2.

Algorithm 2 Naive batch selection

procedure NAIVE-SELECT(C, S, M, b)
  $R \leftarrow \emptyset$
  while $|R| \neq b$ **do**
    $\widetilde{C} = C \setminus R$
    $p^* = \arg\max_{p \in \widetilde{C}} S(M, p)$
    $R \leftarrow R \cup p^*$
  end while
  return R
end procedure

Surrogate Believer Batch Selection. This strategy is inspired by the Kriging Believer strategy [23, 24]. It takes advantage of the scoring function's reliance on the uncertainty of the surrogate model. The main idea is that temporarily adding a point and its model-predicted response value to refit (update) the surrogate model will not drastically alter the response surface, but it will reduce the variance in the area surrounding this point. As both scoring functions used are variance-based, re-scoring the candidates based on the updated surrogate model would likely cause the new highest scoring candidate to be farther away from the last selected point. Formally, this process is described in Algorithm 3, where $M(x)$ returns the predicted response value using the surrogate model at point(s) x.

Topology-Based Batch Selection: Points with Maximum Persistence. We now utilize the topological method described earlier to study the topology of the scoring function. Scoring function S is a real-valued function that takes as input a surrogate model M and a point x, and returns a real

Algorithm 3 Believer batch selection

procedure BELIEVER-SELECT(C, S, M, b)
  $R \leftarrow \emptyset$
  while $|R| \neq b$ **do**
    $\widetilde{C} = C \setminus R$
    $\widetilde{M} \leftarrow M + [R, M(R)]$
    ▷ Build a temporary surrogate model that treats R and $M(R)$ as part of the training set
    $p^* = \arg\max_{p \in \widetilde{C}} S(\widetilde{M}, p)$
    $R \leftarrow R \cup p^*$
  end while
  return R
end procedure

number based on properties of M, the predicted value $M(x)$, the variance of the prediction, etc. When $b = 1$, the naive strategy always select the highest scoring candidate, which topologically, corresponds to the global maximum of the scoring function. When $b > 1$, the naive strategy selects b points with the highest scores, and does not guarantee well-spacing among the selected candidates since these points may cluster around the global maxima. If we would like to guarantee some level of spatial separation among the chosen candidates, then perhaps a simple topology-based strategy would select the top b *local maxima* of the scoring function. However, as the example illustrated in Figure 2, a 1D scoring function could contain two local maxima that are close to each another in the domain but have very different *persistence*. Such an observation has inspired our first topology-based batch selection strategy, based on the persistence of the candidate points. Since regular points have zero persistence, such a strategy selects points among the local maxima. That is, we select the top b candidates with the highest persistence. This strategy is detailed in Algorithm 4.

Algorithm 4 Maximum persistence batch selection

procedure MAXP-SELECT(C, S, M, b)
  $R \leftarrow \emptyset$
  $X \leftarrow$ **EXTRACT-LOCAL-MAXIMA**(C, S, M)
  ▷ Extract all local maxima from the scoring function of the surrogate model
  while $|X| < b$ **do**
    ▷ While loop is only used if we do not have b unselected local maxima
    $\widetilde{C} = C \setminus X$
    $p^* = \arg\max_{p \in \widetilde{C}} S(M, p)$
    $X \leftarrow X \cup p^*$
  end while
  $R \leftarrow R \cup$ **TOP**(X, b) ▷ Get only the top b maxima (according to persistence)
  return R
end procedure

Since we would like to generalize our batch selection process to any arbitrary dimension, we compute an approximated Morse-Smale complex [22] of the scoring function to estimate the location of its local maxima and their corresponding persistence.

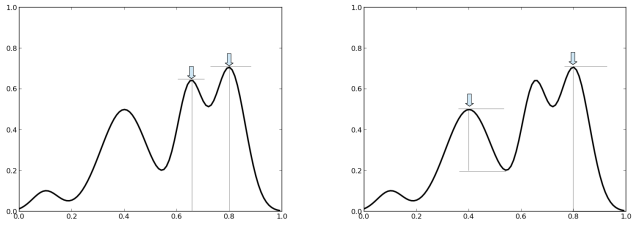

Figure 2: A simple 1D function with 4 maxima showing a selection strategy that chooses the 2 highest-valued maxima (left) and the strategy we employ in MaxP which selects the 2 highest persistence maxima.

Topology-Based Batch Selection: Maximum Persistence and Believer Hybrid. Our final strategy creates a hybrid between the maximum persistence strategy and the Surrogate Believer strategy. That is, we proceed with the believer strategy by temporarily adding points with the highest persistence. We introduce one extra parameter m that indicates how many points we add temporarily at

the same time. Equivalently, m means the number of maxima to chose before refitting the surrogate. When $m = 1$, this corresponds to the original believer strategy. When $m = b$, this corresponds to the maximum persistence strategy. When $1 < m < b$, we have our hybrid strategy. The process is detailed in Algorithm 5.

Algorithm 5 Maximum persistence and Believer hybrid selection

procedure MAXP+BELIEVE-SELECT(C, S, M, b, (optional) m)
  $R \leftarrow \emptyset$
  while $|R| \neq b$ **do**
    $\widetilde{C} = C \setminus R$
    $\widetilde{M} \leftarrow M \cup [R, M(R)]$
    $X \leftarrow$ **EXTRACT-LOCAL-MAXIMA**($\widetilde{C}$, S, $\widetilde{M}$)
    ▷ Extract all local maxima from the scoring function of the temporary surrogate model
    $X \leftarrow X \setminus R$
    ▷ Remove any local maxima that have already been selected
    while $|X| < m$ **do**
      ▷ While loop is only used if we do not have m unselected local maxima
      $\widetilde{C} = \widetilde{C} \setminus X$
      $p^* = \arg\max_{p \in \widetilde{C}} S(\widetilde{M}, p)$
      $X \leftarrow X \cup p^*$
    end while
    $R \leftarrow R \cup$ **TOP**(X, m) ▷ Get only the top m maxima (according to persistence)
  end while
  return R
end procedure

5 Experiments and Results

We compare our active learning strategies for both global surface recovery and limit surface recovery problems. We use a collection of 2D test functions and measure the minimum number of sample points required (to be added to the training set) for the surrogate model to reach a chosen approximation quality with respect to the ground truth, measured by some metric. For each test function, we report the mean, median, and standard deviation of this number across 10 trails with randomly selected initial training data.

5.1 Metrics for Evaluation

For the global surface recovery problem, we compute the *root-mean-square error* (RMSE) over a validation set V of the domain between the surrogate model response surface and the true response surface. The validation set V for our 2D test functions is a uniform grid of resolution 100×100.

For the limit surface recovery problem, we use two different metrics to try to quantify the quality of the approximations. The *Hausdorff distance* between the true limit surface and the approximated limit surface based on the surrogate model is computed to measure curve similarity. This metric represents the maximum separation of the two limit surfaces represented as point set samples X and Y. It is defined as:

$$d_H(X, Y) = \max\left(\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y)\right).$$

In addition, we treat our model as a binary classifier with the limit surface being the boundary and compute the F_1-*score*. The F_1-*score* demonstrates the surrogate model's ability to correctly classify points as being either above or below the threshold value. It is evaluated on the validation grid of resolution 100×100.

The Hausdorff distance focuses on the approximation quality of the limit surface itself. Consider a small component of the limit surface that is yet to be recovered, the F1-score may still report a high-value (corresponding to good approximation quality) since the missing component encloses only a small area of the domain space, but the Hausdorff distance may be very large (corresponding to a low approximation quality) as not all components in the limit surface have been recovered. In other words, the Hausdorff distance metric complements the F1-score metric, as the former is a shape-matching criteria and the latter focuses on classification accuracy.

5.2 Test Functions

We first use a collection of 2D synthetic functions (with closed-form expressions) as test functions. As these functions are simple, smooth and continuous, the surrogate models on them converge very quickly during the active learning process. We demonstrate that our methods remain competitive on these functions, where intuitively they are not expected to perform as well since the main features will be recovered quickly and will benefit more from exploitation rather than exploration. In addition, these functions have well-defined, smooth limit surfaces that could be recovered accurately and efficiently, ensuring good convergence qualities of all methods. We include three test functions here, we omit their close-form expressions but illustrate the smooth functions (true response surfaces) in Figure 3(a)-(c). Their corresponding limit surfaces (marked by yellow curves) are shown in Figure 3(d)-(f). The **GMM** corresponds to the surface of a 2D Gaussian mixture model. The GMM has three limit surface components with one designed to be smaller and less likely to be found immediately. The limit surface of the Salomon function contains three concentric rings, while the outermost ring is split into four components due to boundary limitations. The Sinusoidal function has a relatively complex limit surface with multiple components, but spans a large portion of the domain.

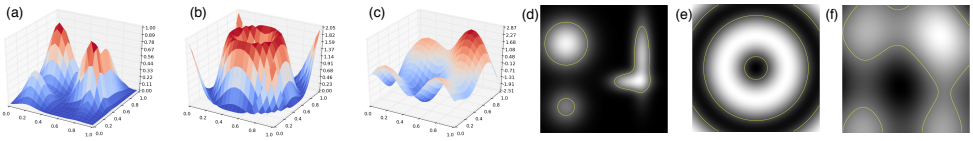

Figure 3: Synthetic test functions and their designated limit surfaces. (a)-(c): True response surfaces for (a) GMM, (b) Salomon and (c) Sinusoidal. (d)-(f): True limit surfaces for (d) GMM, (e) Salomon and (f) Sinusoidal.

To demonstrate that our proposed topology-based active learning methods could potentially be more advantageous with functions that have complex topology, we use a set of greyscale images as our real-world test functions. We treat each image as a 2D function, where its domain is defined by the pixel locations and its range is the luminosity ranging between 0 (black) to 255 (white). The images we have selected contain irregular shapes, discontinuities, and noisy features, which are more indicative of real-world data. They are shown with superimposed limit surfaces in Figure 4. The limit surfaces are highly irregular with many small components, and their corresponding threshold values are 125, 30, 80, and 100, respectively.

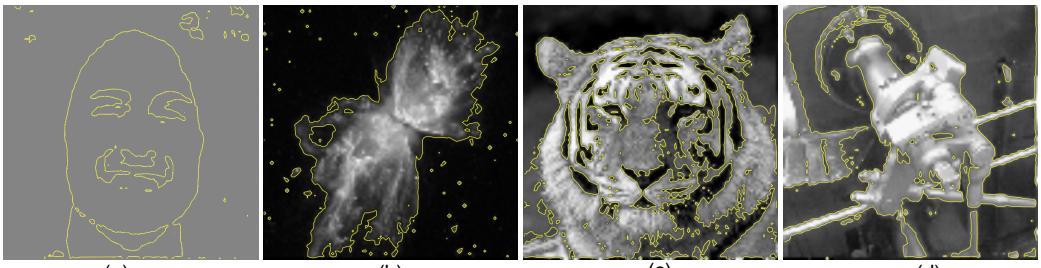

Figure 4: Test images. True limit surfaces are marked with yellow curves. (a) Face (greyscale image removed to preserve author anonymity). (b) Butterfly nebula (image courtesy of NASA, ESA and the Hubble SM4 ERO Team, in public domain, via Wikimedia Commons). (c) Tiger (image courtesy of Wikimedia Commons). (d) Valve (image courtesy of Wikimedia Commons).

5.3 Results for Global Surface Recovery

RMSE Convergence Results. Table 1 summarizes the number of points necessary to reduce the global RMSE under a given threshold for each test function. The threshold value is chosen by examining the RMSE convergence across all trials, e.g. we stop the active learning process when RMSE falls below 0.07818 for the GMM dataset. We report the mean number of points required as well as the standard deviation for each strategy across 10 trials. The maximum number of points added for the synthetic test functions is capped at 256 and for the image datasets, at 1024. We

experiment with three different batch sizes, for b equals to 8, 16 and 32, respectively. The strategy names are pretty self-explainatory, for example, 2-MaxP+Believe means the hybrid strategy where the parameter $m = 2$. The strategy with the best performance for each dataset is highlighted in bold font. If the best strategy is topology-based, it is colored in red.

	GMM ($\leq$ 0.07818)	Salomon ($\leq$ 0.07169)	Sinusoidal ($\leq$ 0.04674)
($b = 8$)			
Naive	120.8 $\pm$ 28.72 (108)	152.8 $\pm$ 13.12 (152)	132.0 $\pm$ 8.198 (128)
Believe	52.8 $\pm$ 9.6 (52)	**91.2 $\pm$ 3.919 (88)**	**68.0 $\pm$ 4.0 (68)**
MaxP	56.0 $\pm$ 8.0 (56)	93.6 $\pm$ 3.666 (96)	75.2 $\pm$ 5.307 (76)
2-MaxP + Believe	**52.0 $\pm$ 9.633 (48)**	93.6 $\pm$ 5.122 (96)	68.8 $\pm$ 3.919 (72)
4-MaxP + Believe	56.8 $\pm$ 15.37 (52)	94.4 $\pm$ 3.2 (96)	70.4 $\pm$ 4.8 (72)
($b = 16$)			
Naive	169.6 $\pm$ 36.63 (160)	203.2 $\pm$ 12.5 (200)	190.4 $\pm$ 16.7 (192)
Believe	59.2 $\pm$ 14.4 (56)	**96.0 $\pm$ 0 (96)**	**72.0 $\pm$ 8.0 (72)**
MaxP	60.8 $\pm$ 9.6 (64)	105.6 $\pm$ 7.838 (112)	84.8 $\pm$ 7.332 (80)
2-MaxP + Believe	**57.6 $\pm$ 14.66 (48)**	97.6 $\pm$ 4.8 (96)	73.6 $\pm$ 7.838 (80)
4-MaxP + Believe	60.8 $\pm$ 15.68 (56)	**96.0 $\pm$ 0 (96)**	75.2 $\pm$ 7.332 (80)
($b = 32$)			
Naive	192.0 $\pm$ 28.62 (192)	243.2 $\pm$ 15.68 (256)	249.6 $\pm$ 12.8 (256)
Believe	**67.2 $\pm$ 9.6 (64)**	**96.0 $\pm$ 0 (96)**	**80.0 $\pm$ 16.0 (80)**
MaxP	99.2 $\pm$ 17.23 (96)	140.8 $\pm$ 15.68 (128)	124.8 $\pm$ 9.6 (128)
2-MaxP + Believe	**67.2 $\pm$ 9.6 (64)**	99.2 $\pm$ 9.6 (96)	83.2 $\pm$ 15.68 (96)
4-MaxP + Believe	70.4 $\pm$ 12.8 (64)	**96.0 $\pm$ 0 (96)**	86.4 $\pm$ 14.66 (96)

Table 1: RMSE convergence results for synthetic test functions.

Increasing the batch size b decreases the rate of convergence, since refitting the surrogate more frequently with small batch sizes would likely select more informative samples. The naive method typically gives a baseline performance that is much worse than all other methods, but it shows the range of performance values and illustrates that the Believe, MaxP, and their hybrid strategies are competitive with one another. On these smooth, relatively simple surfaces, the results shown here slightly favor the Believe strategy, in most cases. The MaxP strategy degrades poorly when increasing the batch size from 16 to 32, most likely due to the fact that there are likely less than 32 local maxima extracted from the ALM scoring function. When all the local maxima have been selected, the remaining candidates are chosen based inn the naive method. By coupling the MaxP strategy with the Believe strategy, we see relatively good performances across many scenarios, since such a hybrid strategy takes advantage of exploiting the topology of the scoring functions.

	Face ($\leq$ 29.83)	Nebula ($\leq$ 19.8)	Tiger ($\leq$ 51.94)	Valve ($\leq$ 38.52)
($b = 8$)				
Naive	776.0 $\pm$ 69.47 (768)	770.4 $\pm$ 96.8 (748)	766.4 $\pm$ 48.77 (768)	683.2 $\pm$ 71.3 (680)
Believe	**755.2 $\pm$ 56.02 (772)**	626.4 $\pm$ 112.7 (640)	732.0 $\pm$ 97.6 (728)	619.2 $\pm$ 99.1 (604)
MaxP	796.0 $\pm$ 51.75 (800)	624.0 $\pm$ 109.5 (596)	**721.6 $\pm$ 125.9 (704)**	628.8 $\pm$ 72.99 (632)
2-MaxP + Believe	749.6 $\pm$ 54.73 (744)	680.0 $\pm$ 49.19 (676)	792.8 $\pm$ 113.1 (816)	**581.6 $\pm$ 97.33 (588)**
4-MaxP + Believe	756.8 $\pm$ 50.87 (764)	**603.2 $\pm$ 106.7 (616)**	777.6 $\pm$ 90.85 (780)	608.8 $\pm$ 94.42 (620)
($b = 16$)				
Naive	832.0 $\pm$ 91.63 (816)	872.0 $\pm$ 98.95 (864)	772.8 $\pm$ 121.4 (792)	756.8 $\pm$ 58.15 (760)
Believe	**756.8 $\pm$ 58.15 (776)**	630.4 $\pm$ 111.1 (640)	**732.8 $\pm$ 90.45 (712)**	630.4 $\pm$ 98.16 (632)
MaxP	820.8 $\pm$ 20.3 (816)	656.0 $\pm$ 115.2 (648)	737.6 $\pm$ 149.3 (800)	595.2 $\pm$ 87.87 (576)
2-MaxP + Believe	779.2 $\pm$ 70.85 (768)	688.0 $\pm$ 66.74 (680)	788.8 $\pm$ 109.5 (784)	600.0 $\pm$ 93.91 (592)
4-MaxP + Believe	768.0 $\pm$ 73.67 (776)	**617.6 $\pm$ 85.33 (680)**	737.6 $\pm$ 97.97 (736)	**590.4 $\pm$ 98.49 (592)**
($b = 32$)				
Naive	918.4 $\pm$ 84.72 (944)	953.6 $\pm$ 69.82 (992)	825.6 $\pm$ 112.5 (800)	838.4 $\pm$ 86.81 (848)
Believe	768.0 $\pm$ 68.63 (784)	**608.0 $\pm$ 108.0 (640)**	**761.6 $\pm$ 99.97 (768)**	640.0 $\pm$ 90.51 (640)
MaxP	828.8 $\pm$ 36.35 (832)	742.4 $\pm$ 104.0 (736)	832.0 $\pm$ 91.63 (832)	**617.6 $\pm$ 111.8 (576)**
2-MaxP + Believe	787.2 $\pm$ 32.63 (784)	739.2 $\pm$ 103.6 (704)	806.4 $\pm$ 129.4 (784)	**617.6 $\pm$ 116.3 (608)**
4-MaxP + Believe	**761.6 $\pm$ 49.16 (768)**	656.0 $\pm$ 90.79 (608)	803.2 $\pm$ 84.0 (800)	630.4 $\pm$ 79.74 (640)

Table 2: RMSE convergence results for image datasets.

The results for the image datasets are shown in Table 2. These datasets exhibit more complicated convergence behaviors. They require more relaxed RMSE convergence thresholds and a larger number of samples to reach modest levels of approximations. In the case of the butterfly nebula image and the tiger image when $b = 8$, the MaxP method outperforms the Believe strategy. In many other cases, the hybrid strategies have the best performance in terms of mean values.

5.4 Results for Limit Surface Recovery

We discuss results for limit surface recovery in this section. First, to understand the difference between the believe strategy and the topology-based MaxP strategy, we show some snapshots of the predicted response surfaces, as well as the corresponding scoring functions (i.e. straddle) during the active learning process in Figure 5, when 152 candidates have already been added to the original 100 training points. The yellow contours denote the predicted limit surface. The blue points are the training data (the same for both trials) and the red points are the adaptively selected points. The green points represent the next batch of 8 selected points. The most important observation is that the Believe strategy can still select points clustered in a single region, potentially limiting the amount of information gain from a single batch, as illustrated by point enclosed in the white circles in Figure 5(a)-(b). This is due in part to the straddle function encoding both uncertainty (in the form of variance) and closeness between the predicted value and the limiting surface threshold. Recall the advantage of an adaptive sampling strategy over a space-filling design is that samples can be sequentially chosen in a proportionately small area of interest. However, we emphasize that our purpose of using batch selection is to explore different areas simultaneously. We believe our topology-based designs are more advantageous in such a context, as illustrated in Figure 5(c)-(d) where the (green) points selected are well-spaced from one another, capturing structural information from different areas of the domain.

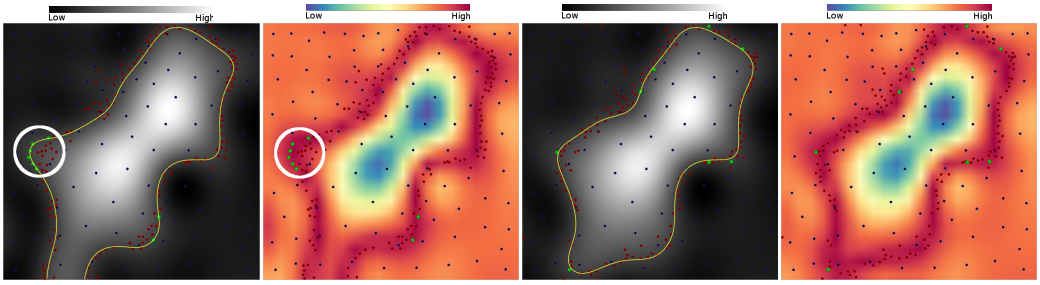

Figure 5: (a)-(b): (a) Predicted response surface based on surrogate model using the Believe strategy and (b) its corresponding scoring function. (c)-(d): (c) Predicted response surface based on MaxP strategy and (d) its corresponding scoring function.

F1-Score Convergence Results. Treating the limit surface recovery problem as a binary classification, the F1-score can be used to understand our model's ability to correctly classify points. Table 3 presents the convergence results. For these synthetic functions, the majority of tests favor the Believe strategy. This is most likely due to the fact that on such simple and smooth examples, there is little need to explore the global space after a few rounds of sampling, and exploiting around the identified limit surface areas is more advantageous. As the topology-based methods are believed to be more judicious about not placing samples very close to one another, they will not converge as quickly as the pure Believe strategy, which can select clusters of points.

The results for image datasets (Table 4) indicates that utilizing the topology of the scoring function in the active learning process is advantageous. With a more complicated response surface, the corresponding scoring function also has complicated topology, which is believed to favor topology-based techniques. For a fixed dataset (e.g. Face, Nebula), the topology-based methods could perform better than the Believe strategy as b creases. This is because the topology of the scoring function is complex enough that even after 32 samples, we can still find additional candidates that are structurally informative.

Hausdorff Distance Convergence Results. Finally, we report the Hausdorff distance metric convergence results in Table 5, computed between the true limit surfaces and the predicted limit surfaces. We omit the image datasets as the Hausdorff distances did not converge on these tests, most likely due to the many small components of the true limit surfaces. As stated earlier, the Hasudorff distance is a measure of the maximum minimum distance between two datasets, therefore failing to capture even a single small component will cause the reported distance to be fairly large. Thus, this metric is better suited for the synthetic functions that have larger, simple limit surface components.

	GMM (≤ 0.9871)	Salomon (≤ 0.988)	Sinusoidal (≤ 0.9942)
($b = 8$)			
Naive	114.4 ± 12.42 (112)	158.4 ± 16.7 (156)	148.0 ± 26.59 (140)
Believe	**60.0 ± 9.633 (64)**	**120.8 ± 14.51 (120)**	**104.0 ± 13.86 (96)**
MaxP	69.6 ± 10.15 (64)	142.4 ± 18.52 (140)	116.8 ± 18.66 (116)
2-MaxP + Believe	**60.0 ± 8.198 (60)**	142.4 ± 14.22 (148)	116.0 ± 19.02 (112)
4-MaxP + Believe	67.2 ± 5.307 (68)	144.0 ± 14.31 (144)	125.6 ± 19.61 (128)
($b = 16$)			
Naive	168.0 ± 21.76 (168)	195.2 ± 15.68 (192)	176.0 ± 25.8 (176)
Believe	67.2 ± 6.4 (64)	**126.4 ± 15.09 (128)**	**107.2 ± 26.82 (96)**
MaxP	81.6 ± 11.2 (80)	137.6 ± 10.61 (136)	116.8 ± 17.6 (112)
2-MaxP + Believe	**64.0 ± 7.155 (64)**	129.6 ± 8.616 (128)	108.8 ± 17.23 (112)
4-MaxP + Believe	70.4 ± 7.838 (64)	137.6 ± 16.32 (128)	116.8 ± 14.4 (112)
($b = 32$)			
Naive	224.0 ± 32.0 (224)	243.2 ± 15.68 (256)	208.0 ± 25.8 (192)
Believe	**80.0 ± 16.0 (80)**	140.8 ± 15.68 (128)	**131.2 ± 33.41 (128)**
MaxP	118.4 ± 28.8 (112)	169.6 ± 14.66 (160)	144.0 ± 21.47 (128)
2-MaxP + Believe	86.4 ± 20.49 (96)	147.2 ± 15.68 (160)	**131.2 ± 22.4 (128)**
4-MaxP + Believe	86.4 ± 14.66 (96)	**134.4 ± 12.8 (128)**	128.0 ± 14.31 (128)

Table 3: F1-Score convergence results for synthetic datasets.

	Face (≤ 0.9281)	Nebula (≤ 0.9387)	Tiger (≤ 0.6933)	Valve (≤ 0.8547)
($b = 8$)				
Naive	**38.4 ± 14.22 (32)**	110.4 ± 39.65 (120)	149.6 ± 103.5 (116)	165.6 ± 81.04 (184)
Believe	41.6 ± 19.2 (32)	107.2 ± 39.22 (112)	144.8 ± 89.33 (112)	**132.0 ± 58.38 (144)**
MaxP	60.0 ± 15.7 (60)	**95.2 ± 39.43 (104)**	**143.2 ± 100.0 (124)**	153.6 ± 82.97 (136*)
2-MaxP + Believe	57.6 ± 16.7 (56)	100.0 ± 39.56 (112)	192.0 ± 198.7 (124)	**132.8 ± 59.24 (144)**
4-MaxP + Believe	58.4 ± 16.41 (60)	97.6 ± 40.6 (96*)	174.4 ± 202.2 (84*)	169.6 ± 91.83 (164)
($b = 16$)				
Naive	**48.0 ± 18.93 (48)**	121.6 ± 44.22 (128)	168.0 ± 134.1 (104*)	139.2 ± 85.58 (136*)
Believe	49.6 ± 19.53 (48)	128.0 ± 45.25 (136)	163.2 ± 135.9 (104*)	**137.6 ± 96.32 (152)**
MaxP	52.8 ± 17.6 (48)	112.0 ± 42.33 (120)	160.0 ± 89.94 (152)	180.8 ± 116.7 (152)
2-MaxP + Believe	54.4 ± 16.32 (48)	**108.8 ± 39.06 (120)**	**153.6 ± 93.35 (144)**	161.6 ± 104.3 (160)
4-MaxP + Believe	52.8 ± 17.6 (48)	**108.8 ± 39.06 (120)**	155.2 ± 89.1 (136)	155.2 ± 99.16 (160)
($b = 32$)				
Naive	67.2 ± 17.23 (64)	150.4 ± 47.57 (160)	195.2 ± 111.3 (176)	240.0 ± 150.9 (224)
Believe	67.2 ± 17.23 (64)	140.8 ± 40.98 (160)	208.0 ± 117.4 (192)	233.6 ± 137.3 (224)
MaxP	64.0 ± 14.31 (64)	121.6 ± 39.97 (128)	169.6 ± 129.6 (128*)	**160.0 ± 72.97 (160)**
2-MaxP + Believe	**60.8 ± 17.23 (64)**	**118.4 ± 38.0 (128)**	**166.4 ± 103.0 (144)**	198.4 ± 104.0 (192)
4-MaxP + Believe	**60.8 ± 17.23 (64)**	**118.4 ± 38.0 (128)**	169.6 ± 103.2 (160)	198.4 ± 104.0 (192)

Table 4: F1-Score convergence results for image datasets. *Denotes where the fastest median convergence differs from the fastest mean convergence.

The Hausdorff distance metric shows even for the more simple test cases that the topology-based methods perform relatively well. This is most likely due to the fact that topology-based methods tend to select well-spaced candidates while still target interesting regions. Spreading candidate points along the estimated limit surface is more likely to reduce the Hausdorff distance computation. Selecting points that are bundled in one area may improve the local fit, but are not likely to reduce the maximum distance computed.

References

[1] Javad Azimi, Alan Fern, and Xiaoli Z. Fern. Batch Bayesian Optimization via Simulation Matching. In *Advances in Neural Information Processing Systems*, pages 109–117, 2010.

[2] Javad Azimi, Alan Fern, Xiaoli Zhang-Fern, Glencora Borradaile, and Brent Heeringa. Batch Active Learning via Coordinated Matching. June 2012.

[3] Javad Azimi, Ali Jalali, and Xiaoli Fern. Hybrid Batch Bayesian Optimization. February 2012.

[4] B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland. Efficient Global Reliability Analysis for Nonlinear Implicit Performance Functions. *AIAA Journal*, 46(10):2459–2468, October 2008.

[5] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 1. Springer New York, 2006.

	GMM (≤ 0.1878)	Salomon (≤ 0.03332)	Sinusoidal (≤ 0.1213)
($b = 8$)			
Naive	84.8 $\pm$ 25.6 (88)	117.6 $\pm$ 44.26 (108)	37.6 $\pm$ 9.499 (40)
Believe	**40.0 $\pm$ 10.12 (36)**	61.6 $\pm$ 27.02 (56)	20.0 $\pm$ 4.0 (20)
MaxP	**40.8 $\pm$ 12.62 (40)**	**48.0 $\pm$ 17.16 (44)**	19.2 $\pm$ 6.4 (16)
2-MaxP + Believe	**40.0 $\pm$ 7.155 (40)**	75.2 $\pm$ 33.98 (68)	18.4 $\pm$ 5.122 (16)
4-MaxP + Believe	44.8 $\pm$ 19.98 (36)	53.6 $\pm$ 16.41 (56)	**17.6 $\pm$ 5.987 (16)**
($b = 16$)			
Naive	113.6 $\pm$ 30.73 (104)	132.8 $\pm$ 30.4 (136)	51.2 $\pm$ 13.95 (56)
Believe	48.0 $\pm$ 10.12 (48)	67.2 $\pm$ 26.58 (64)	19.2 $\pm$ 6.4 (16)
MaxP	**44.8 $\pm$ 21.23 (40)**	**52.8 $\pm$ 17.6 (48)**	22.4 $\pm$ 7.838 (16)
2-MaxP + Believe	46.4 $\pm$ 11.2 (48)	72.0 $\pm$ 27.01 (64)	**17.6 $\pm$ 4.8 (16)**
4-MaxP + Believe	48.0 $\pm$ 16.0 (48)	56.0 $\pm$ 16.4 (56)	22.4 $\pm$ 7.838 (16)
($b = 32$)			
Naive	169.6 $\pm$ 43.05 (160)	169.6 $\pm$ 47.57 (160)	76.8 $\pm$ 15.68 (64)
Believe	64.0 $\pm$ 20.24 (64)	99.2 $\pm$ 26.58 (96)	**32.0 $\pm$ 0 (32)**
MaxP	67.2 $\pm$ 30.19 (64)	**83.2 $\pm$ 21.23 (80)**	35.2 $\pm$ 9.6 (32)
2-MaxP + Believe	60.8 $\pm$ 22.4 (64)	99.2 $\pm$ 30.19 (96)	**32.0 $\pm$ 0 (32)**
4-MaxP + Believe	**57.6 $\pm$ 19.2 (64)**	102.4 $\pm$ 34.47 (96)	**32.0 $\pm$ 0 (32)**

Table 5: Hausdorff distance convergence results for synthetic datasets.

[6] Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *In Proceedings of the 20th International Conference on Machine Learning*, pages 59–66. AAAI Press, 2003.

[7] Brent Bryan. *Actively Learning Specific Function Properties with Applications to Statistical Inference*. PhD thesis, School of Computer Science, Dec 2007.

[8] Brent Bryan, Jeff Schneider, Robert C. Nichol, Christopher J. Miller, Christopher R. Genovese, and Larry Wasserman. Active learning for identifying function threshold boundaries. NIPS, 2005.

[9] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[10] Carlos D. Correa and Peter Lindstrom. Towards robust topology of sparsely sampled data. *IEEE Transactions on Visualization and Computer Graphics*, 17:1852–1861, 2011.

[11] Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural Comput.*, 14(3):641–668, March 2002.

[12] Thomas Desautels, Andreas Krause, and Joel Burdick. Parallelizing Exploration-Exploitation Tradeoffs with Gaussian Process Bandit Optimization. June 2012.

[13] Animesh Dey and Sankaran Mahadevan. Ductile structural system reliability analysis using adaptive importance sampling. *Structural Safety*, 20(2):137–154, January 1998.

[14] Harris Drucker, Chris JC Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. *Advances in neural information processing systems*, pages 155–161, 1997.

[15] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Rev.*, 41(4):637–676, 1999.

[16] Herbert Edelsbrunner, John Harer, Vijay Natarajan, and Valerio Pascucci. Morse-Smale complexes for piecewise linear 3-manifolds. *Proceedings 19th ACM Symposium on Computational Geometry*, pages 361–370, 2003.

[17] Herbert Edelsbrunner, John Harer, and Afra J. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete and Computational Geometry*, 30:87–107, 2003.

[18] Herbert Edelsbrunner, David Letscher, and Afra J. Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28:511–533, 2002.

[19] M. S. Eldred, H. Agarwal, V. M. Perez, S. F. Wojtkiewicz, and J. E. Renaud. Investigation of reliability method formulations in DAKOTA/UQ. *Structure and Infrastructure Engineering*, 3(3):199–213, September 2007.

[20] Michael Eldred and Barron Bichon. *Second-Order Reliability Formulations in DAKOTA/UQ*. American Institute of Aeronautics and Astronautics, 2014/07/11 2006.

[21] Roman Garnett, Yamuna Krishnamurthy, Xuehan Xiong, Jeff Schneider, and Richard P Mann. Bayesian optimal active search and surveying. In J. Langrod and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, pages 1239—1246, Madison, WI, USA, 2012. Omnipress.

[22] Samuel Gerber, Peer-Timo Bremer, Valerio Pascucci, and Ross T. Whitaker. Visual exploration of high dimensional scalar functions. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1271–1280, 2010.

[23] David Ginsbourger, Rodolphe Le Riche, Laurent Carraro, and Département Mi. A multi-points criterion for deterministic parallel global optimization based on gaussian processes. *Journal of Global Optimization, in revision*, 2009.

[24] Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause. Active learning for level set estimation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.

[25] Yuhong Guo. Active instance sampling via matrix partition. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 802–810. Curran Associates, Inc., 2010.

[26] Yuhong Guo and Dale Schuurmans. Discriminative batch mode active learning.

[27] Attila Gyulassy, Vijay Natarajan, Valerio Pascucci, and Bernd Hamann. Efficient computation of Morse-Smale complexes for three-dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 13:1440–1447, 2007.

[28] Steven C. H. Hoi, Rong Jin, and Michael R. Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 633–642, New York, NY, USA, 2006. ACM.

[29] Steven C. H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 417–424, New York, NY, USA, 2006. ACM.

[30] R.L. Iman, J.M. Davenport, and D.K. Zeigler. *Latin hypercube sampling (program user's guide). [LHC, in FORTRAN]*. Jan 1980.

[31] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 13(4):455–492, December 1998.

[32] C.Q. Lam and The Ohio State University. *Sequential Adaptive Designs in Computer Experiments for Response Surface Model Fit*. The Ohio State University, 2008.

[33] David J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

[34] David J.C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[35] Dan Maljovec, Bei Wang, Ana Kupresanin, Gardar Johannesson, Valerio Pascucci, and Peer-Timo Bremer. Adaptive sampling with topological scores. *International Journal for Uncertainty Quantification*, 3(2):119–141, 2013.

[36] Dan Maljovec, Bei Wang, Diego Mandelli, Peer-Timo Bremer, and Valerio Pascucci. Adaptive sampling algorithms for probabilistic risk assessment of nuclear simulations. *International Topical Meeting on Probabilistic Safety Assessment and Analysis (PSA)*, 2013.

[37] Dan Maljovec, Bei Wang, Diego Mandelli, Peer-Timo Bremer, and Valerio Pascucci. Analyzing dynamic probabilistic risk assessment data through clustering. *International Topical Meeting on Probabilistic Safety Assessment and Analysis (PSA)*, 2013.

[38] Andrew W. Moore and Jeff Schneider. Memory-based stochastic optimization. In *Neural Information Processing Systems 8*, pages 1066–1072. MIT Press, 1996.

[39] Valerio Pascucci, Giorgio Scorzelli, Peer-Timo Bremer, and Ajith Mascarenhas. Robust on-line computation of reeb graphs: simplicity and speed. *ACM Transactions on Graphics*, 26(3):58, 2007.

[40] P. Ranjan, D. Bingham, and G. Michailidis. Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, 50(4):527–541, 2008.

[41] Carl Edward Rasmussen. Gaussian processes for machine learning. MIT Press, 2006.

[42] G. Reeb. 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.

[43] Warren S. Sarle. Neural networks and statistical models, 1994.

[44] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[45] AlexJ. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[46] Donald F. Specht. A general regression neural network. *Neural Networks, IEEE Transactions on*, 2(6):568–576, 1991.

[47] Johan AK Suykens, Jos De Brabanter, Lukas Lukas, and Joos Vandewalle. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, 48(1):85–105, 2002.

[48] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[49] S. Tong and D. Koller. Active learning for parameter estimation in Bayesian networks. In *Conference on Advances in Neural Infomation Processing Systems (NIPS 2000)*, Denver, Colorado, 2001.

[50] Y.-T. Wu. Adaptive importance sampling (ais)-based system reliability sensitivity analysis method. In PolD. Spanos and Y.-T. Wu, editors, *Probabilistic Structural Mechanics: Advances in Structural Reliability Methods*, International Union of Theoretical and Applied Mechanics, pages 550–564. Springer Berlin Heidelberg, 1994.

[51] Zuobing Xu, Ram Akella, and Yi Zhang. Incorporating diversity and density in active learning for relevance feedback.

[52] T. Zou, S. Mahadevan, Z. Mourelatos, and P. Meernik. Reliability analysis of automotive body-door subsystem. *Reliability Engineering & System Safety*, 78(3):315–324, December 2002.