12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM
17 - 19 September 2012, Indianapolis, Indiana

AIAA 2012-5437

14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 17-19 September 2012, Indianapolis, Indiana

Uncertainty Quantification in Surrogate Models Based on Pattern Classification of Cross-validation Errors

Jie Zhang* and Souma Chowdhury*
Rensselaer Polytechnic Institute, Troy, New York 12180

Ali Mehmani† and Achille Messac‡
Syracuse University, Syracuse, NY, 13244

This paper advances the Domain Segmentation based on Uncertainty in the Surrogate (DSUS) framework which is a novel approach to characterize the uncertainty in surrogates. The leave-one-out cross-validation technique is adopted in the DSUS framework to measure local errors of a surrogate. A method is proposed in this paper to evaluate the performance of the leave-out-out cross-validation errors as local error measures. This method evaluates local errors by comparing: (i) the leave-one-out cross-validation error with (ii) the actual local error estimated within a local hypercube for each training point. The comparison results show that the leave-one-out cross-validation strategy can capture the local errors of a surrogate. The DSUS framework is then applied to key aspects of wind resource assessment and wind farm cost modeling. The uncertainties in the wind farm cost and the wind power potential are successfully characterized, which provides designers/users more confidence when using these models.

Keywords: Cross-validation, pattern classification, support vector machine, surrogate modeling, uncertainty, wind farm cost, wind resource assessment

I. Introduction

Since a surrogate model is an approximation to an unknown function, prediction errors are generally present in the estimated function values. The two major sources of uncertainty in surrogate modeling are: (i) uncertainty in the observations (when they are noisy), and (ii) uncertainty due to finite sample. One of the major challenges in surrogate modeling is to accurately quantify these uncertainties, and how these uncertainties vary in the design space.[1] It would be uniquely helpful to be able to quantify what level of uncertainty is expected in the surrogate model. Additionally, the knowledge of how the surrogate uncertainty and the levels of errors vary in the variable space will introduce more confidence in the usage of the surrogate (irrespective of its overall level of fidelity).

Can we segregate the input space of a surrogate, based on the accuracy of the surrogate in different regions, and characterize the uncertainty in each region? By addressing this question, we can quantify the uncertainty in the surrogate during the training process itself, which is applicable to a majority of surrogate models.

*Doctoral Student, Multidisciplinary Design and Optimization Laboratory, Department of Mechanical, Aerospace and Nuclear Engineering, AIAA student member

†Doctoral Student, Multidisciplinary Design and Optimization Laboratory, Department of Mechanical and Aerospace Engineering, AIAA student member

‡Distinguished Professor and Department Chair. Department of Mechanical and Aerospace Engineering, AIAA Lifetime Fellow. Corresponding author. Email: messac@syr.edu

A. Uncertainty in Surrogate Modeling

Uncertainties can generally be classified into two categories: (i) aleatoric, or statistical, uncertainties; and (ii) epistemic, or systematic, uncertainties. Epistemic uncertainty represents a lack of knowledge about the appropriate value to use for a quantity, which can be reduced through increased or more relevant data. This paper focuses on the epistemic uncertainty to quantify prediction uncertainties in surrogate models.

Uncertainty and error quantification is a classical theme in surrogate modeling. In surrogate modeling, the uncertainty arises from not knowing the output of the simulation code, except at a finite set of training points. Apley et al.[2] referred to this type of uncertainty as "interpolation uncertainty". In this paper, we use the term "prediction uncertainty", because we believe that surrogate modeling is one type of predictive models.

Bayesian methods[2,3] are widely used to quantify the interpolation uncertainty in computer experiments. Kennedy and O'Hagan[3] developed a Bayesian approach to calibrate a computer code by using observations from the real process, and subsequent prediction and uncertainty analysis of the process. The unknown inputs are represented as a parameter vector θ. Using the training data, Kennedy and O'Hagan derived the posterior distribution of θ, which in particular quantifies the uncertainty about θ. Apley et al.[2] developed an approach using Bayesian prediction intervals (PIs) for quantifying the effects of interpolation uncertainty on the objective function in robust design. Closed-form analytical expressions were derived for the prediction intervals on the response mean, the response variance, and the robust design objective function.

Neufeld et al.[4] assessed the uncertainty introduced by a surrogate model in the conceptual design of the wing box of a generic light jet, by applying Reliability Based Design Optimization (RBDO) to obtain a feasible solution. A Kriging surrogate was developed from a database of finite element analysis solutions sampled across 200 evenly distributed design points. The ratios between the finite element method-based and the predicted stress were represented by a normal distribution.

Picheny[1] showed that uncertainty in conservative predictions can be compensated by adding bias to the surrogate models in order to increase safety, using constant or pointwise margins. The margins are designed based on the error distribution measures given by the model (Kriging), or based on the model-independent accuracy measures (cross-validation) of the model.

Surrogate-based optimization under uncertainty has been conducted in the literature.[5–9] Uncertainty estimates are used in adaptive sampling and optimization methods to select the next sampling point(s). The Efficient Global Optimization (EGO) approach[6] and the Sequential Kriging Optimization (SKO) algorithm[10] use the Kriging uncertainty to seek the point of maximum expected improvement as the next infill point. Viana and Haftka[7] proposed the importation of uncertainty estimates from one surrogate to another. A Support Vector Regression (SVR) with an uncertainty model was developed by combining the prediction from SVR and the standard error from Kriging. Xiong et al.[8] developed cheap surrogate models to integrate information from both low-fidelity and high-fidelity models based on the Bayesian-Gaussian process modeling. The interpolation uncertainty of the surrogate model due to the lack of sufficient high-fidelity simulations is quantified in the Bayesian-Gaussian process modeling.

The existing uncertainty in surrogates modeling methods are model-dependent. A generalized methodology that can be applied to a majority of surrogate models to characterize the uncertainty in surrogates will be more helpful. To this end, we developed an innovative approach to segregate the design domain based on levels of fidelity, and quantify the uncertainty in the surrogate model development itself.[11]

B. Research Objectives

Surrogate models can be used with more confidence if we can do two things: (i) quantify the uncertainty in the surrogate model, and (ii) characterize how the levels of errors vary in the variable space. A new methodology was recently developed to characterize the variation of errors over the design domain, called Domain Segmentation based on Uncertainty in the Surrogate (DSUS).[11] This technique segregates the design domain based on the level of cross-validation errors. The estimated errors are classified into physically meaningful classes based on the user's understanding of the system and/or the accuracy requirements for the concerned system analysis. In each class, the distribution of the cross-validation errors is estimated to represent the uncertainty in the surrogate.

In the DSUS framework, the leave-one-out cross-validation errors at all training points, in conjunction with the RAE metric, are used as local error measures. The objectives of this paper include: (i) evaluating the performance of cross-validation errors as local error measures; and (ii) applying the DSUS framework to

key aspects of wind resource assessment and wind farm cost modeling.

The remainder of the paper is organized as follows: Section II briefly presents the DSUS framework; the use of cross-validation errors as local error measures is evaluated in Section III; Section IV applied the DSUS method to a series of problems in wind resource assessment and wind farm cost modeling; and the numerical settings and results of case studies are shown and discussed in Section V.

II. Domain Segmentation based on Uncertainty in the Surrogate (DSUS)

A brief description of the DSUS framework is provided in this section. The uncertainty in each class is defined by the distribution of leave-one-out cross-validation errors. It is important to note that, in this paper, the term "prediction uncertainty" denotes the distribution of errors of the surrogate model prediction.

A. The Overall DSUS Framework

For an engineering design model, it is important to balance the accuracy of model predictions with the computational complexity of the model used. Model prediction errors that lie within a specified range may be acceptable. Based on the current level of knowledge regarding the design problem, the designer may know what levels of errors are acceptable for particular design purposes. For instance, for a wind farm power generation model, less than 5 percent estimation error might be desirable; 5-10 percent error is acceptable; and higher than 10 percent error is unlikely acceptable. If the whole design domain can be divided into physically meaningful classes based on the level of prediction errors, new designs can be classified into a given class. Figure 1 illustrates the concept of determining the predictive modeling errors in a two design variable system. In Fig. 1, the errors of the model are classified into three classes, and each color represents one class of errors. The boundaries between classes can be determined using pattern classification methods. The designer can estimate the confidence of a new design based on the region into which the design point is classified; these regions can correspond to "good", "acceptable", and "unacceptable" levels of accuracy.

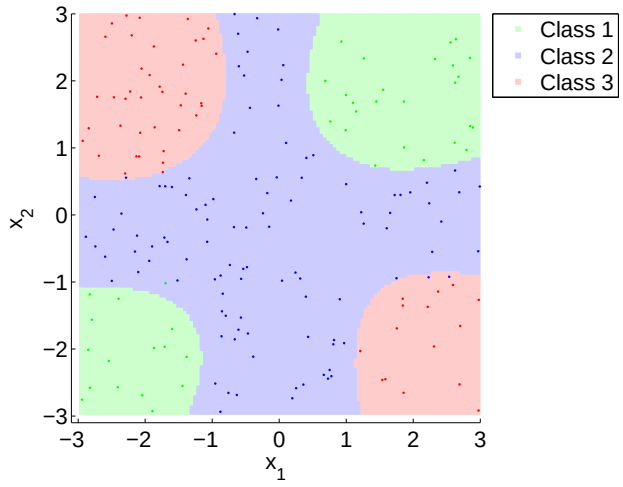


Figure 1. The illustration of the *prediction uncertainty* modeling (class 1: low error; class 2: medium error; class 3: high error)

The DSUS framework is implemented by observing the following sequence of four steps.

1. An interpolating surrogate modeling method is used to develop the predictive model: *e.g.*, Kriging, Radial Basis Functions (RBF), or Adaptive Hybrid Functions (AHF).

2. Cross-validation errors are evaluated using the leave-one-out strategy; and we classify the training points into classes based on the cross-validation error at each training point. Within each class, the distribution of the cross-validation errors is represented using a Gaussian distribution.

3. A model is developed to determine the class boundaries in the input variable space using Support Vector Machines (SVM). The input variables of the surrogates are thus considered as input features in

the classification process.

4. Stochastic models are developed (using Gaussian probability density functions) to represent the surrogate uncertainty in each error class.

It is important to note that in steps 1 and 2, approximating surrogates or regression models (*e.g.*, PRSM) can also be used. However, in that case, a direct error estimation at each training point may be used instead of cross-validation errors. The development of the DSUS framework is illustrated in Fig. 2. In the following sections, we discuss the components of the DSUS framework.

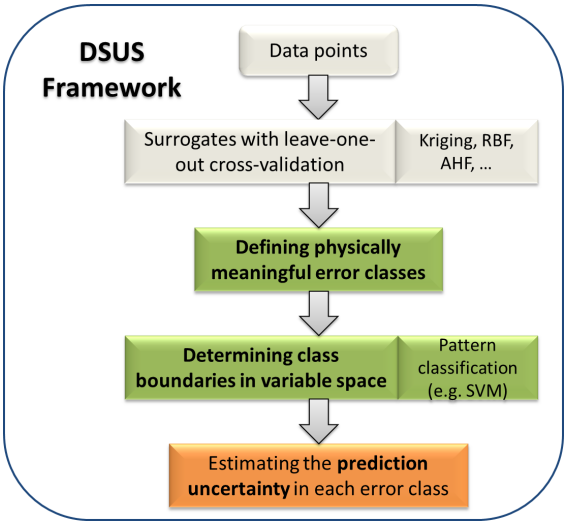


Figure 2. The framework of the DSUS methodology

B. Cross-Validation Errors

Cross-validation errors are used in this paper as a measure of the accuracy of the surrogate. Cross-validation is a technique that is used to analyze and improve the accuracy of a surrogate model. Cross-validation error is the error estimated at a data point, when the response surface is fitted to all the data points except that point (also called the leave-one-out strategy). A vector of cross-validation errors, $\tilde{e}$, can be obtained, when the response surfaces are fitted to all the other points. This vector is known as the prediction sum of squares (the PRESS vector).

In the case of problems where the surrogate modeling process has a significant time expense, the leave-one-out strategy can be computationally expensive for a large number of training points. This issue can be overcome by using the q-fold strategy. Q-fold strategy involves (i) splitting the data randomly into q (approximately) equal subsets, (ii) removing each of these subsets in turn, and (iii) fitting the model to the remaining $q-1$ subsets. A loss function L can be computed to measure the error between the predictor and the points in the subset that we set aside at each iteration; the contributions to L are then summed up over the q iterations.

In order to obtain the error at each training point, the leave-one-out strategy is adopted in the DSUS framework. The Relative Accuracy Error (RAE) is used to classify the training points into classes. The RAE is evaluated at each training point as

$$RAE(X^k) = \left| \frac{\tilde{f}_c(X^k) - f(X^k)}{f(X^k)} \right|, \qquad k = 1, 2, \cdots, n_{tr} \tag{1}$$

where n_{tr} is the number of training points; $f(x^k)$ represents the actual function value at the training point x^k; and the term $\tilde{f}_c(x^k)$ is the corresponding function value at x^k, estimated by the surrogate. That is developed using all training points except x^k, using the leave-one-out cross-validation strategy. According to the RAE values, we can manually classify the training points into error classes, and define the lower and upper limits of each class. The definition of the error levels (error ranges) depends on how the user intends

to use the domain-based surrogate uncertainty information, *e.g.*, use it for increasing the confidence of a particular system analysis.

C. Pattern Classification

The classes generated in the previous step are used to determine the classification boundaries. A wide variety of pattern classification methods are available in the literature,[12] such as (i) linear discriminant analysis (LDA); (ii) principal components analysis (PCA); (iii) kernel estimation and K-nearest-neighbor algorithms; (iv) Perceptrons; (v) neural network; and (vi) Support Vector Machine (SVM). In this paper, the uncertainty classification in surrogate modeling is a multiclass classification problem. Support Vector Machine (SVM), which has been reported to be a competitive approach for multiclass classification problem,[13] is adopted in this paper.

1. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a popular machine learning technique that has been used for classification, regression, and other learning tasks. Given a training set of instance-label pairs $(x_i, y_i),\ i = 1, \cdots, m$ where $x_i \in R^n$ and $y \in \{1, -1\}^m$, the determination of the support vectors requires the solution of the following optimization problem:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i \\ \text{subject to} \quad & y_i \left(w^T \phi(x_i) + b \right) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{aligned} \tag{2}$$

Here, the generic training vector x_i is mapped onto a higher (maybe infinite) dimensional space by the function ϕ. SVM finds a hyperplane with the maximum margin in this higher dimensional space. The vector w denotes the normal vector to the hyperplane; and the parameter $\frac{b}{\|w\|}$ determines the offset of the hyperplane from the origin along the normal vector w. The parameter $C > 0$ is the penalty parameter in the error term. The generic term ξ is a slack variable, which measures the degree of misclassification of the datum x_i. The function, $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is called the kernel function. Four kernels that are popularly used are:

1. Linear: $K(x_i, x_j) = x_i^T x_j$
2. Polynomial: $K(x_i, x_j) = \left(\gamma x_i^T x_j + r \right)^d,\ \gamma > 0$
3. Radial basis function: $K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2 \right),\ \gamma > 0$
4. Sigmoid: $K(x_i, x_j) = tanh\left(\gamma x_i^T x_j + r \right)$

where γ, r, and d are kernel parameters.

The SVM is a powerful tool for binary classification, which is capable of generating fast classifier functions following a training period. There exist several advanced SVM approaches to solve classification problems that involves three or more classes:[14]

1. One-against-all classification, in which one binary SVM is used to separate members of each class from the members of other classes.
2. One-against-one classification, which constructs $k(k-1)/2$ classifiers where each classifier is trained on data from two classes (k is the number of classes).
3. Directed acyclic graph SVM (DAGSVM), in which the training phase is the same as the one-against-one method.

Hsu and Lin[14] provided a detailed comparison of the above three approaches and concluded that "one-against-one" classification is a competitive approach. Thus, the one-against-one approach is adopted in this

paper. For training data from the i^{th} and the j^{th} classes, the following two-class classification problem is solved:[15]

$$\begin{aligned} \min_{w^{ij},b^{ij},\xi^{ij}} \quad & \frac{1}{2}\left(w^{ij}\right)^T w + C\sum_t \left(\xi^{ij}\right)_t \\ \text{subject to} \quad & \\ & \left(w^{ij}\right)^T \phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \quad \text{if } x_t \text{ in the } i^{th} \text{ class} \\ & \left(w^{ij}\right)^T \phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \quad \text{if } x_t \text{ in the } j^{th} \text{ class} \\ & \xi_t^{ij} \geq 0 \end{aligned} \tag{3}$$

A voting strategy is used in classification. Each binary classification is considered to be a voting process where votes can be cast for all data points x; and in the end a point is designated to be in the class with the maximum number of votes.[15] In this paper, we have used an efficient SVM package, LIBSVM (A Library for Support Vector Machines), developed by Chang and Lin.[15]

III. Illustrating Cross-validation Errors as Local Error Measures

Both global and local error measure metrics were used to assess the accuracy of surrogate models. In general, cross-validation (prediction sum of squares) is used as one of the global measure metrics. Local error metrics include Maximum Absolute Error (MAE) and Relative Accuracy Error (RAE). In the DSUS framework, the leave-one-out cross-validation errors at all training points, in conjunction with the RAE metric, are used as local error measures. A method is proposed in this paper to evaluate the performance of this local error measure metric.

The DSUS framework can be applied in conjunction with a majority of standard surrogate modeling methods. A newly developed hybrid surrogate, the Adaptive Hybrid Functions (AHF),[16] is adopted in this paper.

A. Adaptive Hybrid Functions (AHF)

The AHF methodology, recently developed by Zhang et al.,[16,17] formulates a reliable trust region, and adaptively combines characteristically different surrogate models. The weight of each contributing surrogate model is represented as a function of the input domain, based on a local *measure of accuracy* of that surrogate model. Such an approach exploits the advantages of each component surrogate, thereby capturing both the global and the local trend of complex functional relationships. In this paper, the AHF combines three component surrogate models by characterizing and evaluating the local *measure of accuracy* of each model. The three models are (i) RBF, (ii) Extended Radial Basis Functions (E-RBF), and (iii) Kriging. The development of the AHF surrogate model is illustrated in Fig. 3. The details of the AHF method can be found in the paper by Zhang et al.[16]

B. Surrogate Prediction in The Neighborhood of Training Points

The local errors of the surrogate are evaluated in the neighborhood of each training point. A local hypercube is constructed to include one training point; and the local accuracy of the surrogate is estimated with the local hypercube. To construct the local hypercube, for a n_d dimension training point $X^j = (x_1^j, x_2^j, \cdots, x_{n_d}^j)$, we first find its nearest neighbor X^{j-n} with the minimum Euclidean distance. The length of the hypercube along each dimension is determined as follows.

$$L_i = \left| x_i^j - x_i^{j-n} \right|, \quad \forall\ i = 1, 2, \cdots, n_d \tag{4}$$

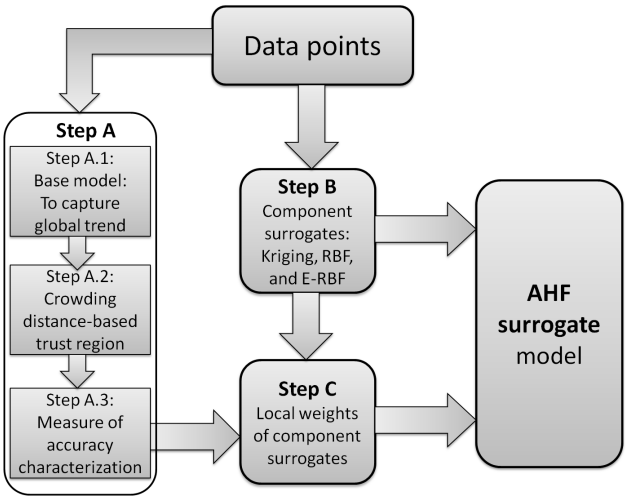


Figure 3. The framework of the AHF surrogate model

The j^{th} hypercube is expressed by

$$H_j = \left\{ (x_1^{jL}, x_1^{jU}), (x_2^{jL}, x_2^{jU}), \cdots, (x_{n_d}^{jL}, x_{n_d}^{jU}) \right\} \tag{5}$$

where

$$x_i^{jL} = x_i^j - \frac{L_i}{2}, \quad \forall\ i = 1, 2, \cdots, n_d$$

$$x_i^{jU} = x_i^j + \frac{U_i}{2}, \quad \forall\ i = 1, 2, \cdots, n_d$$

The parameters x_i^{jL} and x_i^{jU} define the boundaries of the local hypercube. The total number of vertices in the hypercube is equal to 2^{n_d}.

Within each local hypercube, a set of test points are generated. The responses of the test points are estimated using the surrogate trained by all training points. The Relative Accuracy Error (RAE) at each test point within the j^{th} local hypercube is given by

$$RAE^{te}(X^{jk}) = \left| \frac{\tilde{f}(X^{jk}) - f(X^{jk})}{f(X^{jk})} \right|, \quad \forall\ j = 1, 2, \cdots, n_{tr}, \quad k = 1, 2, \cdots, n_{te} \tag{6}$$

where $f(X^{jk})$ represents the actual function value at the test point X^{jk}; $\tilde{f}(X^{jk})$ is the estimated response at the point X^{jk} using the surrogate trained by all training points; and n_{te} is the number of test points within each local hypercube. For the j^{th} local hypercube, the average of the RAE values for the n_{te} points, $\overline{RAE^{te}}_j$, is determined by

$$\overline{RAE^{te}}_j = \frac{1}{n_{te}} \sum_{k=1}^{n_{te}} RAE^{te}(X^{jk}) \tag{7}$$

where $RAE^{te}(X^{jk})$ represents the RAE value for the k^{th} test point within the j^{th} local hypercube.

The local errors estimated by the leave-one-out cross-validation strategy are evaluated at each training point by comparing: (i) the RAE value estimated by the leave-one-out surrogate (Eq. 1); and (ii) the $\overline{RAE^{te}}$ value estimated within the local hypercube (Eq. 7).

C. Analytical Examples

Two analytical examples are used to show the surrogate prediction in the neighborhood of training points, which are: (i) the 2-variable Dixon & Price function, and (ii) the 2-variable Booth function. The expressions of the two functions are given as follows.

Test Function 1: 2-variable Dixon & Price Function

$$f(x) = (x_1 - 1)^2 + 2\left(2x_2^2 - x_1\right)^2 \quad (8)$$
$$where \quad x_i \in [-10 \;\; 10]$$

Test Function 2: 2-variable Booth Function

$$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \quad (9)$$
$$where \quad x_i \in [-10 \;\; 10]$$

For both problems, 30 training points are generated to construct the surrogate. The Latin Hypercube sampling strategy developed by Audze-Eglais[18] is adopted to determine the locations of sample points. The following optimality criterion is used in the sampling method.

$$\sum_{p}^{n_{tr}} \sum_{q=p+1}^{n_{tr}} \frac{1}{L_{pq}^2} \rightarrow min \quad (10)$$

where L_{pq} is the distance between the points p and q, and n_{tr} is the total number of sample points.

The RAE error at each training point and the $\overline{RAE^{te}}$ value for each local hypercuble are estimated. The values of RAE and $\overline{RAE^{te}}$ are classified into 4 groups, based on the mean values of RAE and $\overline{RAE^{te}}$ for all the 30 training points. Figures 4 and 5 show the comparison of cross-validation errors and actual local errors for the Dixon & Price function and the Booth function, respectively. The 4 classes are defined as follows.

$$RAE_i \in \begin{cases} Class\ 1 & if \;\; RAE_i < 0.5\mu_{RAE} \\ Class\ 2 & if \;\; 0.5\mu_{RAE} \leq RAE_i < \mu_{RAE} \\ Class\ 3 & if \;\; \mu_{RAE} \leq RAE_i < 1.5\mu_{RAE} \\ Class\ 4 & if \;\; RAE_i \geq 1.5\mu_{RAE} \end{cases} \quad (11)$$

where RAE_i represents the RAE value at the i^{th} training point; and μ_{RAE} is the mean RAE value for the 30 training points.

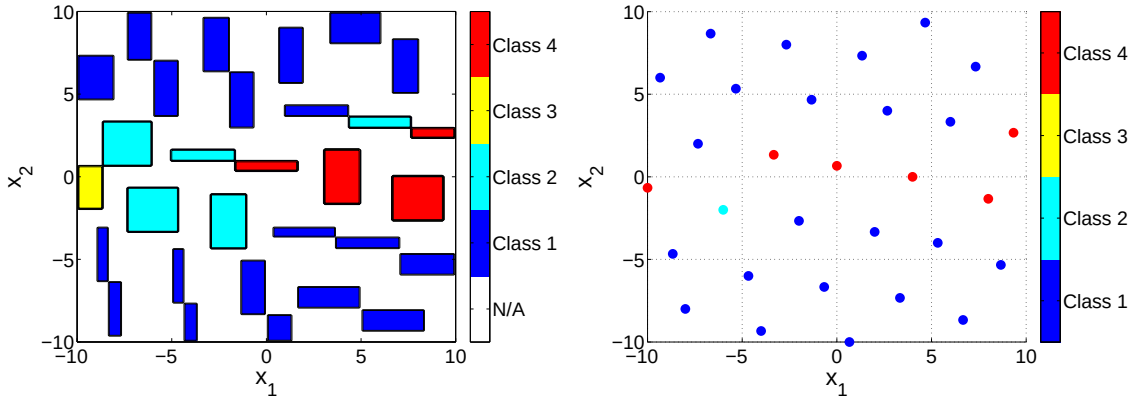
$$\overline{RAE^{te}}_i \in \begin{cases} Class\ 1 & if \;\; \overline{RAE^{te}} < 0.5\mu_{\overline{RAE^{te}}} \\ Class\ 2 & if \;\; 0.5\mu_{\overline{RAE^{te}}} \leq \overline{RAE^{te}} < \mu_{\overline{RAE^{te}}} \\ Class\ 3 & if \;\; \mu_{\overline{RAE^{te}}} \leq \overline{RAE^{te}} < 1.5\mu_{\overline{RAE^{te}}} \\ Class\ 4 & if \;\; \overline{RAE^{te}} \geq 1.5\mu_{\overline{RAE^{te}}} \end{cases} \quad (12)$$

where $\overline{RAE^{te}}_i$ represents the $\overline{RAE^{te}}$ value for the i^{th} local hypercube; and $\mu_{\overline{RAE^{te}}}$ is the mean $\overline{RAE^{te}}$ value for the 30 local hypercubes.

For the test functions, the locations of the training points are illustrated in Figs. 4(b) and 5(b). Figures 4(a) and 5(a) show the local hypercubes at all training points. The levels of errors are represented using different colors. It is observed that: (i) for the Dixon & Price function, the cross-validation errors and the actual local errors belong to the same class for 83.33% of the 30 training points; (ii) for the Booth function, the cross-validation errors and the actual local errors belong to the same class for 86.67% of the 30 training points. The comparison results show that the leave-one-out cross-validation strategy can capture the local errors of a surrogate.

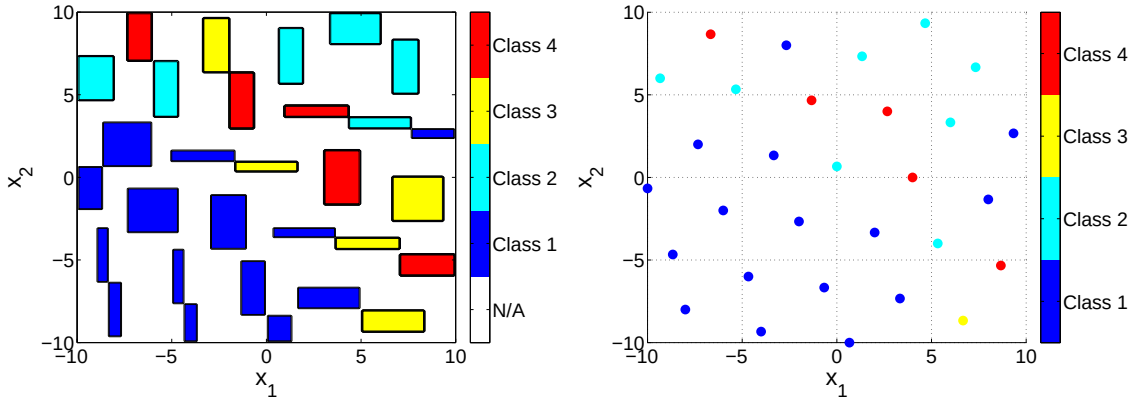
IV. Wind Energy Case Studies

The performance of the new DSUS framework has been illustrated in the previous paper[11] using a set of analytical test problems. In this paper, the DSUS framework is applied to a series of problems in wind resource assessment and wind farm cost modeling, in conjunction with the Adaptive Hybrid Functions (AHF).



(a) Actual local errors (b) Cross-validation errors

Figure 4. Comparison of cross-validation errors and actual local errors (Dixon & Price function)



(a) Actual local errors (b) Cross-validation errors

Figure 5. Comparison of cross-validation errors and actual local errors (Booth function)

A. Onshore Wind Farm Cost Model

A Response Surface-Based Wind Farm Cost (RS-WFC) model was developed by Zhang et al.[19] The RS-WFC model was developed using Extended Radial Basis Functions (E-RBF) for onshore wind farms in the U.S. The RS-WFC model is composed of three components that estimate the effects of engineering and economic factors on (i) the installation cost, (ii) the annual Operation and Maintenance (O&M) cost, and (iii) the total annual cost of a wind farm.

In this paper, we adopted the RS-WFC model to estimate *the total annual cost of a wind farm per kilowatt installed* and to characterize the uncertainty in the cost. The AHF surrogate is used to develop the cost model instead of using the E-RBF method. The input parameters to the RS-WFC total annual cost model are (*i*) the number, and (*ii*) the rated power of each wind turbine installed in the wind farm; and the output is the total annual cost of the wind farm. Data points collected for the state of North Dakota are used to develop the cost model.

B. Wind Power Potential (WPP) Model

The Wind Power Potential (WPP) model[20] is developed to evaluate the wind power potential (maximum wind farm capacity factor for a specified farm size and installed capacity) for differing locations, by observing the following sequence of four steps.

- **STEP 1**: Using recorded wind data at a location, the joint distribution of wind speed and direction is estimated to characterize the wind conditions. Bivariate normal distribution[21] is adopted in this paper. The parameters of bivariate normal distribution include five variables: (i) the mean of the wind speed distribution, μ_U; (ii) the mean of the wind direction distribution, μ_θ; (iii) the variance of the wind speed distribution, σ_U; (iv) the variance of the wind direction distribution, σ_θ; and (v) the correlation coefficient between wind speed and wind direction, ρ. Every unique combination of these five parameters represents a unique sample distribution of wind speed and wind direction.

- **STEP 2**: The five parameters of the bivariate normal distribution are sampled using design of experiment methods. The Sobol's quasirandom sequence generator[22] is adopted in this paper.

- **STEP 3**: For each sample distribution of wind speed and wind direction, we maximize the net power generation through farm layout optimization. To this end, we employ the Unrestricted Wind Farm Layout Optimization (UWFLO) methodology.[23]

- **STEP 4**: A surrogate model is constructed to represent the computed maximum capacity factor as a function of the parameters of the bivariate normal distribution.

For any farm site, according to the recorded wind data, we can (i) estimate the parameters of the joint distribution of wind speed and direction, and (ii) predict the maximum capacity factor for a specified farm size and capacity, using the WPP model. The details of the WPP model can be found in the paper by Zhang et al.[20]

In this paper, we explore two different scenarios in the WPP estimation. The uncertainty in the WPP is characterized using the DSUS framework. We consider a fixed-size (land) rectangular wind farm that is comprised of a defined turbine-type. Two different scenarios are considered here, given as

- Case 1: Evaluating the WPP (maximum capacity factor) of a farm, comprised of four turbines;

- Case 2: Evaluating the WPP (maximum capacity factor) of a farm, comprised of nine turbines.

The *GE-1.5MW-xle*[24] turbine is chosen as the specified turbine-type in both Cases 1 and 2.

V. Results and Discussion

The parameter selections and numerical settings of the DSUS framework are summarized. The SVM kernels and parameters are determined for the studied cases, followed by the discussion of the uncertainty prediction results. The AHF surrogate is an ensemble of RBF, E-RBF and Kriging methods. Detailed formulations of the component surrogates (RBF, E-RBF and Kriging) can be found in the paper by Zhang et al.[16] The parameter values are the same as specified in that paper.

A. Numerical Settings

The numerical settings for each problem are summarized in Table 1, which includes (i) the number of input variables, (ii) the number of training points, (iii) the number of test points, and (iv) the SVM kernel. The data used to develop and test the RS-WFC model is obtained from the Energy Efficiency and Renewable Energy Program at the U.S. Department of Energy.[25] We select the kernel function through numerical experiments; and the radial basis function kernel is adopted for the test problems. For the *Wind Power Potential* (WPP) estimation, the training points are generated using Sobol's quasirandom sequence generator.

Table 1. Numerical setup for test problems

Problem	No. of variables	No. of training points	No. of test points	SVM kernel
Wind farm cost	2	60	20	radial basis function
WPP (4 turbines)	5	100	20	radial basis function
WPP (9 turbines)	5	100	20	radial basis function

We classify the training points into different physically meaningful classes based on the RAE values. The user-defined lower and upper limits of each class are listed in Table 2. It is seen from Table 2 that three classes are defined for the test problems.

Table 2. The uncertainty scale in each class

Problem	Class 1	Class 2	Class 3
Wind farm cost	RAE<0.5%	0.5% $\leq$ RAE<1%	RAE $\geq$ 1%
WPP (4 turbines)	RAE<5%	5% $\leq$ RAE<10%	RAE $\geq$ 10%
WPP (9 turbines)	RAE<5%	5% $\leq$ RAE<10%	RAE $\geq$ 10%

There are two parameters for the SVM radial basis function kernel: C and γ. Cross-validation technique[15] is used to achieving high training accuracy. A grid-search technique[15] is performed on C and γ using cross-validation. Various pairs of (C, γ) values are tested and the one with the best cross-validation accuracy is selected.

B. Uncertainty Prediction Results

1. Uncertainty in The Wind Farm Cost

Table 3 shows the prediction accuracy of each problem. It is seen that the DSUS framework performs fairly well for all the problems. The classification accuracy of the DSUS prediction is more than 90% for all the test problems.

Table 3. Prediction accuracy of each problem

Problem	Parameters	Accuracy
Wind farm cost	C=1, γ=0.8	100% (20/20)
WPP (4 turbines)	C=1, γ=1	90% (18/20)
WPP (9 turbines)	C=1, γ=0.2	95% (19/20)

Table 4 shows the uncertainty in the wind farm cost for three new wind farm designs. For the first new design, the DSUS method classifies the design into error class 1 based on the input parameters of the cost model (40 wind turbines with rated power of 1.25 MW). The uncertainty in the cost of the wind farm is characterized by a Gaussian distribution. The mean and the standard deviation values of the Gaussian distribution are 0.0018 and 0.0011, respectively.

Table 4. Uncertainty characterization for new designs (wind farm cost)

New design	No. of turbines	Rated power	Class	Uncertainty (μ, σ)
1	40	1.25 MW	1	0.0018, 0.0011
2	7	1.00 MW	2	0.0066, 0.0013
3	44	1.50 MW	3	0.0216, 0.0102

2. Uncertainty in The Wind Power Potential

Four locations are selected to evaluate their wind power potentials. The wind data for these four locations is obtained from the *North Dakota Agricultural Weather Network* (NDAWN).[26] The daily averaged data (wind speed and wind direction) in 2010 is measured and recorded at the four differing stations (Ada, Baker, Beach, and Bottineau). The parameters of the bivariate normal distributions for the four stations are estimated using maximum likelihood estimators, which are listed in Table 5.

Table 5. Bivariate normal distribution parameters estimated for the four stations[20]

Station	Mean, μ_U speed	Mean, μ_U direction	Standard deviation, σ_U speed	Standard deviation, σ_θ direction	Correlation, ρ coefficient
Ada	3.89	197.45	1.89	101.04	0.10
Baker	3.94	195.51	1.83	103.74	0.07
Beach	4.00	219.24	1.44	85.97	0.13
Bottineau	3.91	196.38	1.78	103.49	0.15

Table 6 shows the uncertainty in the wind power potential estimations for the four wind farm sites. For the Ada, Baker and Bottineau stations, we observe that errors of the estimated WPP are classified into Class 3 for both the 4-turbine and the 9-turbine wind farms. For the Beach station, the error of the estimated WPP is classified into Class 2 for the 4-turbine wind farm, and is classified into Class 1 for the 9-turbine wind farm. The uncertainty in each error class is represented by a Gaussian distribution. The values of mean and standard deviation are given in Table 6. Classes 1, 2 and 3 represent uncertainty levels of low errors, medium errors and high errors, respectively. The prediction uncertainties in the surrogates for the WPP estimations are illustrated in Fig. 6.

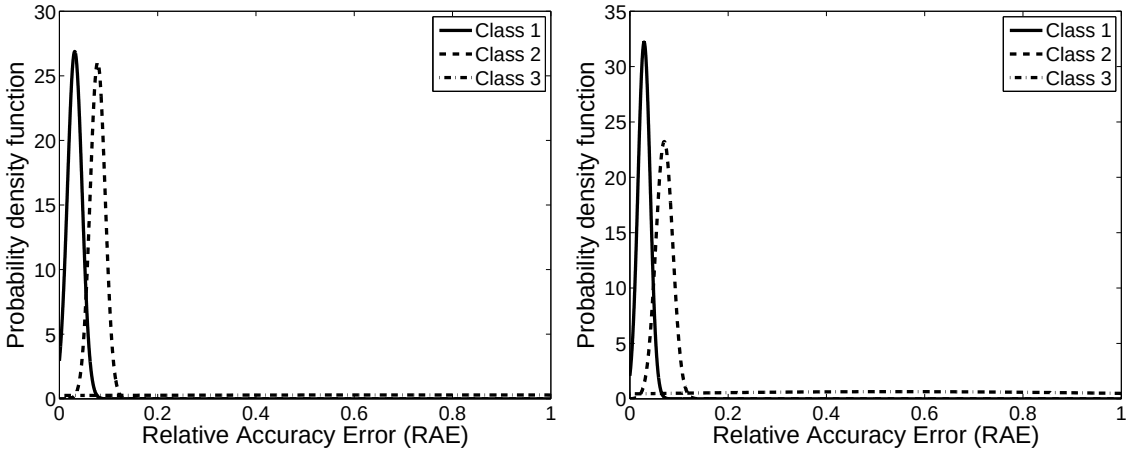
Table 6. Uncertainty in the estimated capacity factors

Station	WPP (4-turbine farm)		WPP (9-turbine farm)	
	Class	Uncertainty (μ, σ)	Class	Uncertainty (μ, σ)
Ada	3	0.7426, 1.3464	3	0.5374, 0.6207
Baker	3	0.7426, 1.3464	3	0.5374, 0.6207
Beach	2	0.0779, 0.0153	1	0.0290, 0.0124
Bottineau	3	0.7426, 1.3464	3	0.5374, 0.6207

VI. Conclusion

The Domain Segmentation based on Uncertainty in the Surrogate (DSUS) framework could successfully characterize the uncertainty attributable to surrogate models. The DSUS framework is useful for a variety of applications, such as optimization, system analysis (involving unknown functional relationships), and surrogate modeling improvement. With the DSUS framework, surrogate modeling techniques can be used with more confidence.

The DSUS framework was applied to wind resource assessment and wind farm cost modeling problems. The uncertainties in the wind farm cost and wind power potential were successfully characterized, which



(a) WPP with 4 turbines

(b) WPP with 9 turbines

Figure 6. Distributions of errors (uncertainy) in each class for the WPP model

provides designers/users more confidence when using these models.

Both global and local error measure metrics were used to assess the accuracy of surrogate models. In the DSUS framework of uncertainty characterization in surrogates, the leave-one-out cross-validation errors at all training points, in conjunction with the RAE metric, are used as local error measures. A method was proposed in this paper to evaluate the performance of the local error measure metric; and the results show that the leave-one-out cross-validation error can capture the local errors of a surrogate. Error measure metrics play important roles in surrogate modeling. However, the error metrics discussed above might not the best way to characterize the local errors in surrogates. In the future, other error metrics that might better represent the performance over the entire design domain should be investigated.

VII. Acknowledgements

Support from the National Science Foundation Awards CMMI-1100948 and CMMI-0946765 is gratefully acknowledged.

References

[1]Picheny, V., *Improving Accuracy and Compensating for Uncertainty in Surrogate Modeling*, Ph.D. thesis, Aerospace Engineering, University of Florida, Gainesville, FL, December 2009.

[2]Apley, D. W., Liu, J., and Chen, W., "Understanding The Effects of Model Uncertainty in Robust Design with Computer Experiments," *Journal of Mechanical Design*, Vol. 128, No. 4, 2006, pp. 945.

[3]Kennedy, M. C. and O'Hagan, A., "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society: Series B*, Vol. 63, No. 3, 2001, pp. 425–464.

[4]Neufeld, D. and an J. Chung, K. B., "Aircraft Wing Box Optimization Considering Uncertainty in Surrogate Models," *Structural and Multidisciplinary Optimization*, Vol. 42, No. 5, 2010, pp. 745–753.

[5]Eldred, M., Giunta, A., Wojtkiewicz, S. F., and Trucano, T., "Formulations for Surrogate-based Optimization Under Uncertainty," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Atlanta, GA, September 4-6 2002.

[6]Jones, D., Schonlau, M., and Welch, W., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, No. 4, 1998, pp. 455–492.

[7]Viana, F. A. C. and Haftka, R. T., "Importing Uncertainty Estimates from One Surrogate to Another," *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA, Palm Springs, California, May 4-6 2009.

[8]Xiong, Y., Chen, W., and Tsui, K., "A New Variable-Fidelity Optimization Framework Based on Model Fusion and Objective-Oriented Sequential Sampling," *Journal of Mechanical Design*, Vol. 130, No. 11, 2008, pp. 111401.

[9]Chen, S., Xiong, Y., and Chen, W., "Multiresponse and Multistage Metamodeling Approach for Design Optimization," *AIAA Journal*, Vol. 47, No. 1, 2009, pp. 206–218.

[10]Huang, D., Allen, T. T., Notz, W. I., and Zeng, N., "Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models," *Journal of Global Optimization*, Vol. 34, No. 3, 2006, pp. 441–466.

[11]Zhang, J., Chowdhury, S., and Messac, A., "Domain Segmentation based on Uncertainty in The Surrogate (DSUS)," *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA, Honolulu, Hawaii, April 23-26 2012.

[12]Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification*, Wiley, New York, 2nd ed., 2000.

[13]Duan, K. and Keerthi, S. S., "Which Is The Best Multiclass SVM Method? An Empirical Study," *Multiple Classifier Systems*, Vol. 3541, 2005, pp. 732–760.

[14]Hsu, C. W. and Lin, C. J., "A Comparison of Methods for Multiclass Support Vector Machines," *IEEE Transactions on Neural Networks*, Vol. 13, No. 2, 2002, pp. 415–425.

[15]Chang, C. C. and Lin, C. J., "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, 2011, pp. 27:1–27:27.

[16]Zhang, J., Chowdhury, S., and Messac, A., "An Adaptive Hybrid Surrogate Model," *Structural and Multidisciplinary Optimization*, Vol. 46, No. 2, 2012, pp. 223–238.

[17]Zhang, J., Chowdhury, S., Messac, A., Zhang, J., and Castillo, L., "Surrogate Modeling of Complex Systems Using Adaptive Hybrid Functions," *ASME 2011 International Design Engineering Technical Conferences (IDETC)*, ASME, Washington, DC, August 28-31 2011.

[18]Audze, P. and Eglais, V., "New Approach for Planning Out of Experiments," *Problems of Dynamics and Strengths*, Vol. 35, 1997, pp. 104–107.

[19]Zhang, J., Chowdhury, S., Messac, A., and Castillo, L., "A Response Surface-based Cost Model for Wind Farm Design," *Energy Policy*, Vol. 42, 2012, pp. 538–550.

[20]Zhang, J., Chowdhury, S., Messac, A., and Castillo, L., "A Comprehensive Measure of The Energy Resource Potential of A Wind Farm Site," *ASME 2011 5th International Conference on Energy Sustainability*, ASME, Washington, DC, August 7-10 2011.

[21]Abramowitz, M. and Stegun, I. A., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Mineola, New York, 1972.

[22]Sobol, I. M., "Uniformly Distributed Sequences with An Additional Uniform Property," *USSR Computational Mathematics and Mathematical Physics*, Vol. 16, No. 5, 1976, pp. 236–242.

[23]Chowdhury, S., Zhang, J., Messac, A., and Castillo, L., "Unrestricted Wind Farm Layout Optimization (UWFLO): Investigating Key Factors Influencing The Maximum Power Generation," *Renewable Energy*, Vol. 38, No. 1, 2012, pp. 16–30.

[24]GE, *GE Energy 1.5MW Wind Turbine Brochure*, General Electric, http://www.gepower.com/, [Accessed: December 2010].

[25]Goldberg, M., *Jobs and Economic Development Impact (JEDI) Model*, National Renewable Energy Laboratory, Golden, Colorado, US, October 2009.

[26]NDSU, "The North Dakota Agricultural Weather Network," http://ndawn.ndsu.nodak.edu, [Accessed: January 2012].