

Research Article

Use of Active Learning to Design Wind Tunnel Runs for Unsteady Cavity Pressure Measurements

Ankur Srivastava and Andrew J. Meade

Mechanical Engineering and Material Science Department, William Marsh Rice University, Houston, TX 77251-1892, USA

Correspondence should be addressed to Ankur Srivastava; ankur.rice@gmail.com

Received 23 December 2013; Accepted 13 May 2014; Published 19 June 2014

Academic Editor: Christopher J. Damaren

Copyright © 2014 A. Srivastava and A. J. Meade. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Wind tunnel tests to measure unsteady cavity flow pressure measurements can be expensive, lengthy, and tedious. In this work, the feasibility of an active machine learning technique to design wind tunnel runs using proxy data is tested. The proposed active learning scheme used scattered data approximation in conjunction with uncertainty sampling (US). We applied the proposed intelligent sampling strategy in characterizing cavity flow classes at subsonic and transonic speeds and demonstrated that the scheme has better classification accuracies, using fewer training points, than a passive Latin Hypercube Sampling (LHS) strategy.

1. Introduction

Internal carriage of stores in aircraft has many aerodynamic advantages especially in military applications. These include enhanced maneuverability, reduced drag, and increased stealth of the aircraft. However, flow over the cavities might generate steady and unsteady disturbances that can affect safe discharge of stores [1]. Figure 1(a) shows the open bomb bay of the F-22 Raptor which can generate self-sustaining oscillations that might lead to cavity resonance risking the structural integrity of the vehicle [2]. Figure 1(b) shows the open missile bay of the F-35 Lightning II that can provide a large nose-up pitching moment to the stores on discharge [3]. These flow disturbances demand extensive computational and experimental studies to be conducted at all operational speeds.

It is widely acknowledged that wind tunnel testing is essential in characterizing flow across a cavity. A number of parameters can influence the flow including freestream Mach number, geometric dimensions of the cavity, and location of stores within the cavity [1, 4, 5]. The number of parameter combinations and requisite data reduction can render wind tunnel testing tedious, expensive, and time consuming. Any innovative mathematical technique that can reduce the time and expense of wind tunnel experiments is welcome.

In a related work, the authors demonstrated how machine learning tools could be used with Design of Experiments (DOEs) to steer the experiment by investigating input parameter sensitivities to the classification of the cavity flow type [6]. The authors used a Galerkin-derived adaptive implementation of artificial neural networks called sequential function approximation (SFA) [6–10] to predict the cavity flow type with or without acoustic resonance as a function of length-to-depth ratio (l/h), width-to-depth ratio (w/h), and the freestream Mach number (M_∞). The authors treated this problem as a multiclass classification problem and justified the selection of SFA by comparison against state-of-the-art classification tools.

However, the work presented in [6] could only steer the cavity experiment by input parameter selection with respect to cavity flow type classification. It could not take part in the data collection procedure given a set of input parameters. In the machine learning literature, such use of a learning algorithm is referred to as passive learning. In this work, we present a learning strategy that actively takes part in the data collection procedure. In other words, we will focus on developing an intelligent sampling strategy that will inform the experimentalist of the most important regions in the test parameter space. Such a strategy that takes part in the data



FIGURE 1: (a) Open bomb bay of a F-22 Raptor [2] and (b) open missile bay of a F-35 Lightning II [3].

sampling procedure is known as active learning [11] in the machine learning community.

1.1. Unsteady Cavity Flow Experiments. Tracy and Plentovich [1] investigated flow characteristics of variable rectangular (Figure 2) cavities in the NASA Langley 8-foot Transonic Pressure Tunnel at subsonic and transonic speeds. They varied the length-to-depth ratio (l/h) from 1 to 17.5, the width-to-depth ratio (w/h) between 1, 4, 8, and 16, and the freestream Mach number (M_∞) from 0.2 to 0.95 at a unit Reynolds number per foot of approximately 3×10^6 .

The cavity flow types were classified into 3 types, namely, open, transitional, and closed. Open flow is associated with deep cavities. The C_p distribution for supersonic cavity flow, shown in Figure 3(a), remains nearly uniform with a sharp increase due to the shock at the trailing edge of the cavity wall. This type of cavity flow is usually associated with self-sustaining oscillations that generates acoustic tones radiating from the cavity. Closed cavity flow usually occurs in shallow cavities where the flow separates both at the forward face and at the rear face of the cavity resulting in a C_p distribution shown in Figure 3(b). The developed adverse static pressure gradient in closed cavities can cause the discharged stores to pitch up [1]. Transitional cavity flow exhibits features of both open and closed flows (Figure 4). In subsonic and transonic flows, a single transitional flow type is defined rather than transitional-open and transitional-closed as in supersonic flows.

Figure 5 summarizes the parameter combinations that support cavity resonance in this experiment. Circles correspond to open flow. Squares correspond to transitional flow. Diamonds correspond to closed flow. Any filled symbol indicates the presence of acoustic resonance. The solid lines illustrate the boundaries between open, transitional, and closed flows.

Tracy and Plentovich [1] conducted wind tunnel runs at 267 different configurations of M_∞ , w/h , and l/h with the primary conclusion that cavities change from resonant to nonresonant conditions as l/h increases. They concluded that the cavity flow type was most sensitive to l/h , while the resonance amplitude and flow transition boundaries varied with w/h . For each parameter combination, labor intensive data reduction was applied to obtain the mean static pressure

distribution and frequency of the pressure oscillations to determine the flow type and the presence of resonance [1].

The primary objective of this work is to explore the use of active machine learning techniques in steering through the wind tunnel test matrix of the Tracy and Plentovich cavity flow pressure measurement problem. We will use uncertainty sampling (US) [12] with SFA as the active learning technique. In this technique, US uses, in a sequential manner, the approximation of the discriminant function constructed by SFA to determine the input configuration that should be chosen to conduct the next wind tunnel run. We will compare generalization ability of SFA when input data is sampled via active learning or by traditional LHS design techniques [13].

In Section 2, we provide a brief introduction to active learning for classification problems. It also presents the details of SFA and its implementation with US, the chosen active learning scheme. Section 3 presents a comparison of prediction accuracies of uncertainty sampling with LHS technique. Section 4 presents conclusions and future work.

2. Approach

2.1. Classification. In statistics, classification of a data set involves the construction of a discriminant function F , which then predicts the labels of the unlabeled data set as accurately as possible. In constructing the discriminant function, one makes the assumption that all of the samples are independent and identically distributed under the same probability distribution function. Unfortunately, this problem is not trivial for a real world data set since the probability distribution from which the samples are drawn is unknown. A number of classification tools, in statistics and function approximation theory, exist in the literature for real world problems and are discussed and compared in detail in [14]. Scattered data function approximation techniques in machine learning are also used to model either the discriminant function directly or the multidimensional manifold that separates the classes (e.g., separating hyper planes from the classic form of Support Vector Machines (SVMs)).

In using scattered data approximation, we assume the discriminant itself is continuous but can only be measured in discrete values (the proxy function). The SFA tool used in this paper attempts to model the underlying function u directly.

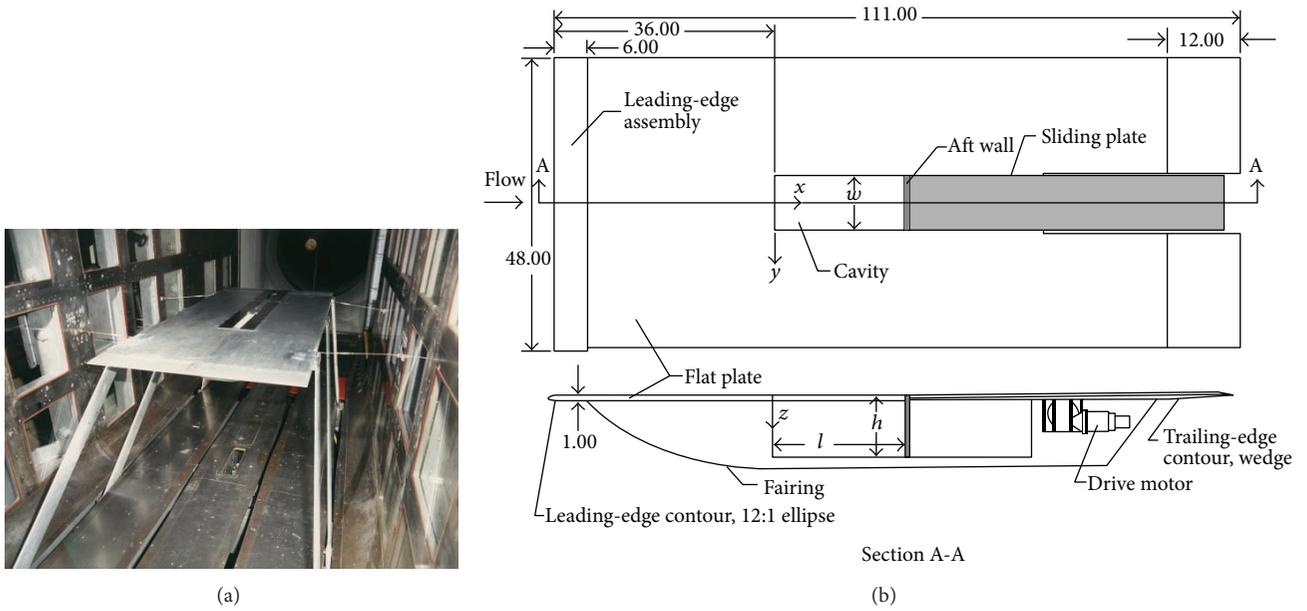


FIGURE 2: Variable cavity wind tunnel model [1]. (a) Downstream view of installation. (b) Sketch of model. Linear dimensions are in inches.

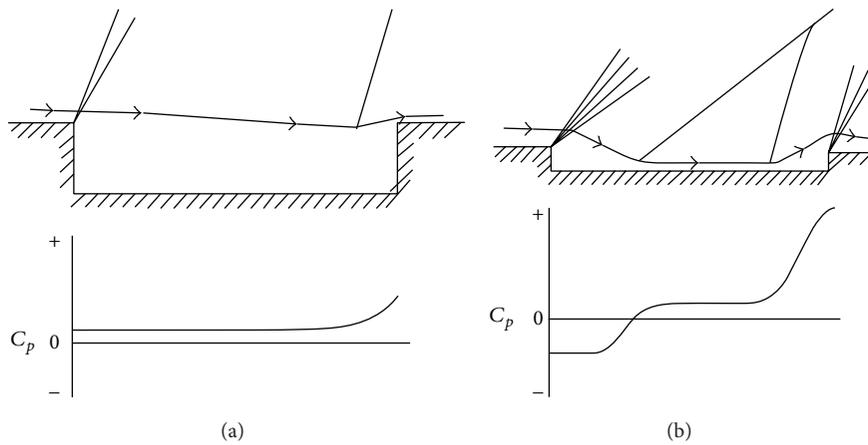


FIGURE 3: (a) Open cavity flow [1] and (b) closed cavity flow [1].

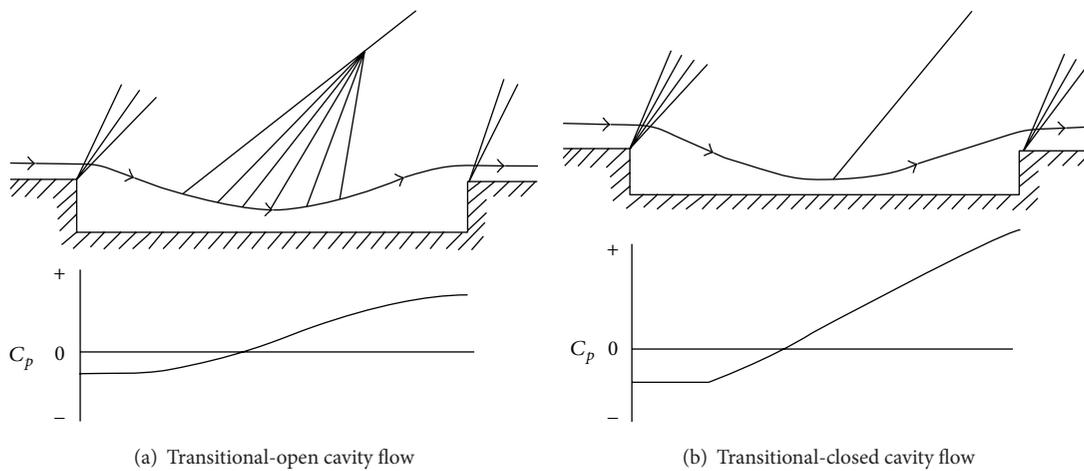


FIGURE 4: Transitional cavity flows [1].

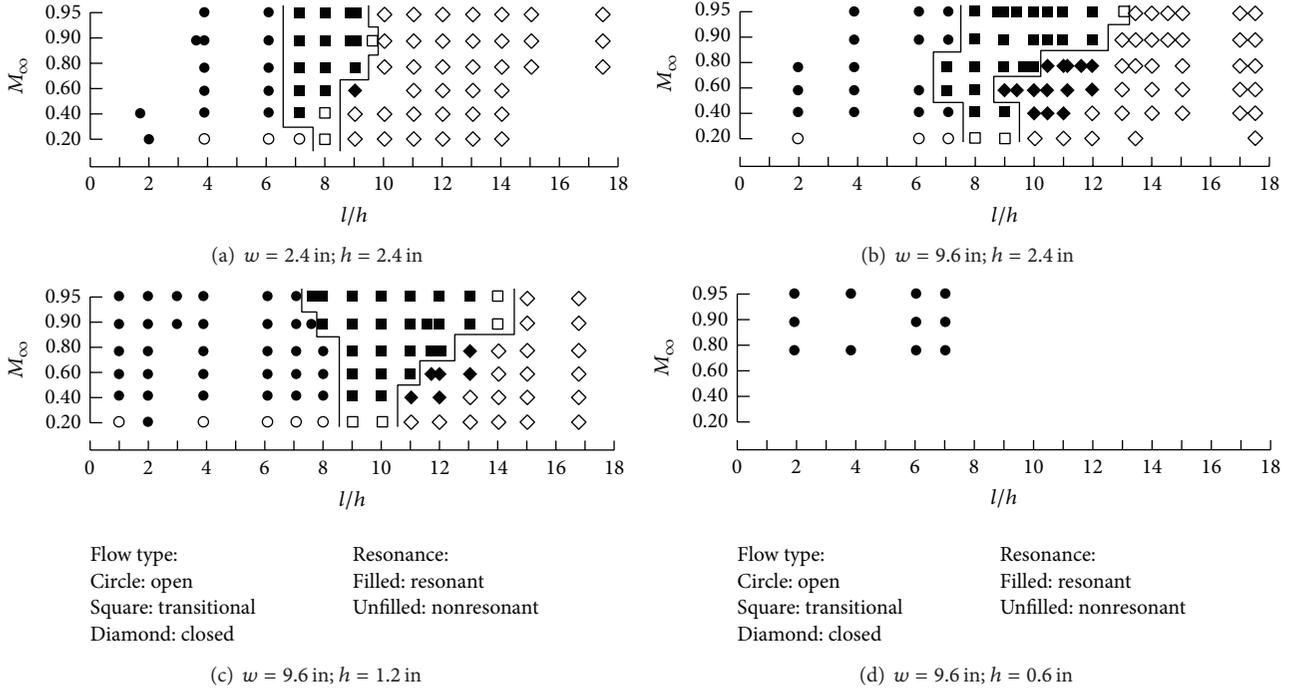


FIGURE 5: Occurrence of resonance superimposed with observed cavity flow conditions [1].

Given a set of d -dimensional input data points $\xi \in R^d$ and noting that a function can be arbitrarily well-approximated by a linear combination of basis functions ϕ , we only require that the approximation fit the scattered data within accepted tolerance; that is,

$$\sum_{i=1}^s \|F[u(\xi_i)] - u_n^a(\xi_i)\|^2 \leq s^2 \tau^2, \quad (1)$$

where $F[u(\xi_i)]$ is the user defined label of the output, $u(\xi_i)$ is the true underlying function, and $u_n^a(\xi_i)$ in (2) is the approximated solution, where

$$u_n^a(\xi) = \sum_{j=1}^n c_j (\phi_j(\xi, \beta_j) + b_j). \quad (2)$$

The values of the coefficients c_j and b_j and the kernel parameters β_j are determined by an optimization routine on the training set. This makes the performance of a classifier dependent on the number and quality of data points available.

Traditionally, Design of Experiment (DOE) methods like the LHS technique are used to obtain input parameter combinations at which experiments are conducted to obtain outputs which are classified by an oracle to obtain their labels. LHS design methods sample data points such that they are uniformly distributed throughout the input space. They do not take into account the functional relationship between the inputs and the output. However, in many real world problems, the output variance is significant in only a limited region of the input domain. In classification problems, output variance is present only at the class discrimination boundaries. LHS designs in these cases might result in sampling data points

from regions which are uninformative. This becomes a significant limitation, especially in problems where either the experiment is costly or the labeling process is tedious or both. To avoid the inefficient use of resources, we focus on sampling the data set utilizing active learning techniques which take into account the input-output functional relationship.

2.2. Active Learning for Classification Problems. In passive learning, as the number of training points of a learning algorithm is increased, the information gained per training sample gradually decreases [11]. To avoid this problem, one can estimate the uncertainty of the learning algorithm on different regions of the input domain and sample data points from the most uncertain regions of the domain. For nonlinear discriminant functions ($F[v]$), data sampling by active learning is essentially sequential in nature. The active learning sampling scheme is as good as the model of $F[v]$ itself. If the model has biases, the chosen data will not be optimal. In real world problems, it is often difficult to determine the true nature of the input-output model (i.e., linear, exponential, and quadratic). However, we can use universal approximators like radial basis functions (RBFs) that automatically address the bias problem because of their smoothness and differentiability. The active learning scheme scans through the input domain to look for regions in which the learning algorithm is least confident. This can be computationally expensive if the number of input dimensions is excessively large. However, this can be addressed through optimum experiment design [15] approaches.

An active learning scheme requires a learning algorithm to construct approximations of the input-output mapping. The desired learning algorithm should be able to construct

a rough approximation of the hypersurface using only a few scattered data points. More importantly, the chosen learning algorithm should be computationally efficient since training and testing will need to be repeated every time a new set of unlabeled samples is added to the training set. This makes the use of grid search and cross-validation approaches for control and kernel parameter optimization computationally infeasible for active learning problems. This is the motivation of the authors in using the self-adaptive and greedy SFA algorithm, with uncertainty sampling, in this work. SFA was developed from mesh-free finite element research but shares similarities with the boosting [16] and matching pursuit [17] algorithms. It was later used to provide kernel based solutions to regression [9] and classification problems [6, 7, 10]. We start our approximation of u utilizing the RBF (ϕ):

$$\begin{aligned} \phi(\xi, \beta_i) &= \exp\left[-\frac{(\xi - \xi_i^*) \cdot (\xi - \xi_i^*)}{\sigma_i^2}\right] \\ &= \exp[\ln[\alpha_i] (\xi - \xi_i^*) \cdot (\xi - \xi_i^*)] \quad \text{for } 0 < \alpha_i < 1, \end{aligned} \quad (3)$$

where $\beta_i = \{\alpha_i, \xi_i^*\} \in \mathbb{R}^{d+1}$ represents the i th basis function parameters, the width α_i , and the center $\xi_i^* \in \mathbb{R}^d$. We write the RBF as (3) in order to set up the optimization problem for α_i as a bounded nonlinear line search instead of an unconstrained minimization problem. The basic principles of our greedy algorithm are motivated by the similarities between the iterative optimization procedures of Jones [18, 19] and Barron [20] and the method of weighted residuals (MWR), specifically the Galerkin method [21]. We can write the function residual r at the n th stage of approximation as in

$$\begin{aligned} r_n &= F[u(\xi)] - u_n^a(\xi) = F[u(\xi)] - u_{n-1}^a(\xi) - c_n(\phi(\xi, \beta_n) + b_n) \\ &= r_{n-1} - c_n(\phi(\xi, \beta_n) + b_n) = r_{n-1} - c_n(\phi_n + b_n). \end{aligned} \quad (4)$$

Using the Petrov-Galerkin approach, we select a coefficient c_n that will force the function residual to be orthogonal to the basis function and b_n using the discrete inner product given by

$$\begin{aligned} &\sum_{j=1}^s r_n(\xi_j) (\phi_n(\xi_j) + b_n) \\ &= \langle r_n, (\phi_n + b_n) \rangle_D = -\left\langle r_n, \frac{\partial r_n}{\partial c_n} \right\rangle_D \\ &= -\frac{1}{2} \frac{\partial \langle r_n, r_n \rangle_D}{\partial c_n} = 0 \end{aligned} \quad (5)$$

which is equivalent to selecting a value of c_n that will minimize $\langle r_n, r_n \rangle_D$ or

$$c_n = \frac{(s \langle \phi_n, r_{n-1} \rangle_D - \langle r_{n-1} \rangle_D \langle \phi_n \rangle_D)}{(s \langle \phi_n, \phi_n \rangle_D - \langle \phi_n \rangle_D \langle \phi_n \rangle_D)} \quad (6)$$

with

$$b_n = \frac{(\langle r_{n-1} \rangle_D \langle \phi_n, \phi_n \rangle_D - \langle \phi_n \rangle_D \langle \phi_n, r_{n-1} \rangle_D)}{(s \langle \phi_n, r_{n-1} \rangle_D - \langle r_{n-1} \rangle_D \langle \phi_n \rangle_D)}. \quad (7)$$

The discrete inner product $\langle r_n, r_n \rangle_D$, which is equivalent to the square of the discrete L_2 norm, can be rewritten, with the substitution of (6) and (7), as

$$\begin{aligned} &\sum_{j=1}^s r_n(\xi_j) r_n(\xi_j) \\ &= \|r_n\|_{2,D}^2 = \langle r_n, r_n \rangle_D \\ &= \left\langle r_{n-1} - \frac{\langle r_{n-1} \rangle_D}{s}, r_{n-1} - \frac{\langle r_{n-1} \rangle_D}{s} \right\rangle_D \\ &\quad \times \left(1 - \left(\left\langle r_{n-1} - \frac{\langle r_{n-1} \rangle_D}{s}, \phi_n - \frac{\langle \phi_n \rangle_D}{s} \right\rangle_D \right)^2 \right. \\ &\quad \times \left(\left\langle \phi_n - \frac{\langle \phi_n \rangle_D}{s}, \phi_n - \frac{\langle \phi_n \rangle_D}{s} \right\rangle_D \right. \\ &\quad \times \left. \left. \left\langle r_{n-1} - \frac{\langle r_{n-1} \rangle_D}{s}, r_{n-1} - \frac{\langle r_{n-1} \rangle_D}{s} \right\rangle_D \right)^{-1} \right). \end{aligned} \quad (8)$$

The dimensionality of the nonlinear optimization problem is kept low since we are solving only one basis at a time. Also, Meade Jr. and Zeldin [22] showed that for bell-shaped basis functions

$$\|r_n\|_2^2 \cong \langle r_n, r_n \rangle_D = \exp[\kappa_1 - \kappa_2 n], \quad \text{where } n \geq 2 \quad (9)$$

for positive constants κ_1 and κ_2 . Equation (9) shows, for optimal convergence, the logarithm of the inner product of the residual as a linear function of the number of bases (n), which establishes an exponential convergence rate that is independent from the number of input dimensions. Though the SFA scheme allows the basis center (ξ_n^*) to be located anywhere in \mathbb{R}^d , the practical application to problems with multiple inputs constrains the centers to the set of sample points $\{\xi_1, \dots, \xi_s\}$. At each stage, we determine ξ_n^* such that $|r_{n-1}(\xi_n^*)| = \max |r_{n-1}|$. The remaining optimization variable α_n is continuous and constrained to $0 < \alpha_n < 1$.

The algorithm terminates when $\max |r_n| \leq \tau$, where τ is the tolerance desired by the user. For binary classification, where $F[u] = +1$ or -1 , we set $\tau = 1.0$ and the approximation output is computed as $\text{sign}[u_n^a(\xi)]$. In this work, our multiclass classification problem is tackled by applying three binary classifiers in parallel. Approaches to combining binary classifiers include the *one versus one* combination, the *one versus all* pairing, and the error correcting output coding. There is no best approach to combining binary classifiers. Even though *one versus all* has its own shortcomings, this approach was selected by the authors because of its simplicity and acceptable accuracy. Rifkin and Klautau [23] compared the *one versus all* approach to other existing methods and concluded that *one versus all* classification will produce

results as good as any other approach if the underlying classifiers are well tuned. In a standard *one versus all* formulation applied to a multiclass classification task, M binary classifiers are constructed where M is the number of classes in the data set. Each classifier predicts the presence of a test point in a particular class in the form of a deterministic real number and the class corresponding to the maximum positive number is assigned to the test point. Say, $Y \in [1, 2, \dots, M]$ and, using our case as an example, $M = 3$, then the class of ξ_{Test} , given by $Y_{\xi_{\text{Test}}}$, would be given by

$$\begin{aligned} Y_{\xi_{\text{Test}}} &= 1 & \text{if } \pi &= u_n^{a1}(\xi_{\text{Test}}), \\ Y_{\xi_{\text{Test}}} &= 2 & \text{if } \pi &= u_n^{a2}(\xi_{\text{Test}}), \\ Y_{\xi_{\text{Test}}} &= 3 & \text{if } \pi &= u_n^{a3}(\xi_{\text{Test}}), \end{aligned} \quad (10)$$

where $\pi = \max(u_n^{a1}(\xi_{\text{Test}}), u_n^{a2}(\xi_{\text{Test}}), u_n^{a3}(\xi_{\text{Test}}))$.

Ties among the classifiers are broken randomly. To implement uncertainty sampling with SFA, the user takes the following steps.

- (1) Choose a class. Label all the members of this class as 1 and the members of all the other classes as -1 .
- (2) Initiate the algorithm with the labels of the training data: $r_0 = \{F[u(\xi_1)], \dots, F[u(\xi_s)]\}$.
- (3) Search the components of r_{n-1} for the maximum magnitude. Record the component index j^* .
- (4) Consider $\xi_{n^*} = \xi_{j^*}$.
- (5) With ϕ_n centered at ξ_{n^*} , minimize (8) to solve for the optimization parameter with α_n .
- (6) Calculate the coefficients c_n and b_n from (6) and (7), respectively.
- (7) Update the residual vector $r_n = r_{n-1} - c_n(\phi_n + b_n)$. Repeat steps 3 to 6 until the termination criterion $\max |r_n| \leq 1.0$ has been met.
- (8) Use the constructed binary classifier to predict the test set.
- (9) Repeat steps 1 to 7 for the remaining classes.
- (10) Use *one versus all* scheme to obtain the final prediction on the test set.

Our binary method is linear in storage with respect to s since it needs to store only $s + sd$ vectors to compute the residuals: one vector of length $s(r_n)$ and s vectors of length $d(\xi_1, \dots, \xi_s)$ where s is the number of samples and d is the number of dimensions. Completing the SFA model requires two vectors of length $n(\{c_1, \dots, c_n\}$ and $\{b_1, \dots, b_n\})$ and n vectors of length $n + 1(\{\alpha_1, \dots, \alpha_n\})$.

2.3. Uncertainty Sampling with SFA. Say we are given a set of d -dimensional input data points ξ and outputs $Y \in [1, 2, \dots, M]$, where M is the number of classes. In binary classification tasks, the output can only take two possible values (-1 or $+1$). A straightforward way to compute the

expected error from the addition of a point ξ to the training set is to calculate

$$E_\xi = P(Y = 1 | \xi) \cdot E_{(\xi, Y=1)} + P(Y = -1 | \xi) \cdot E_{(\xi, Y=-1)}. \quad (11)$$

Equation (11) calculates the overall expected error on a test set when a data point ξ is added to the training set with output $+1$ or -1 . $P(Y = 1 | \xi)$ and $P(Y = -1 | \xi)$ are the posterior probabilities that the label of the training point ξ is $+1$ and -1 , respectively. Even though this approach is straightforward, it is computationally expensive. In binary classification, this computation can be avoided by selecting a data point that lies in the vicinity of the classification boundary. A point lying close to the class discrimination boundary is guaranteed to have an effect on the approximation of the discriminant function. Selecting unlabeled samples that lie in the vicinity of the discriminant boundary falls under the category of uncertainty sampling. Even though US does not optimize an information gain criterion, it has been proven to be effective in many practical applications [24, 25]. Uncertainty sampling bears similarities with the query by committee [26] approach where disagreement between the members of a committee of classifiers is evaluated to choose an unlabeled sample for labeling. The active learning scheme applied to the cavity problem can be schematically represented in Figure 6. Once a final training set has been constructed, the final form of the discriminant function can be constructed to predict labels in real time.

For algorithms like SFA that construct their approximations in the form of (2), points that lie in the vicinity of the classification boundary can be found by looking at the minimum absolute value of the argument of the proxy function. To implement uncertainty sampling with SFA, the following steps need to be followed.

- (1) Use a Design of Experiment method like LHS technique to pick a small number of initial training points to begin the active data collection procedure.
- (2) Conduct training and evaluate the classifier on a pool of unlabeled samples.
- (3) Using u_n^a , determine the test point that bears the minimum absolute value of the argument of the proxy function. Multiple points can also be selected in a similar manner.
- (4) Add the chosen unlabeled samples to the training set and remove them from the pool of the unlabeled samples.
- (5) Repeat steps 2 through 4 until the pool of labeled samples is exhausted.

The application of this heuristic with SFA is demonstrated on the following simulated classification problem given in

$$u(\xi) = \text{sign}[y - x^2] \quad \text{where } 0 \leq x, y \leq 1. \quad (12)$$

The classifier was trained on one-half of the available points and tested on the remainder. Percentage accuracy was calculated by (13) shown below:

$$\text{Percentage accuracy} = \left(\frac{N_{\text{test}} - m_{\text{test}}}{N_{\text{test}}} \right) \times 100\%, \quad (13)$$

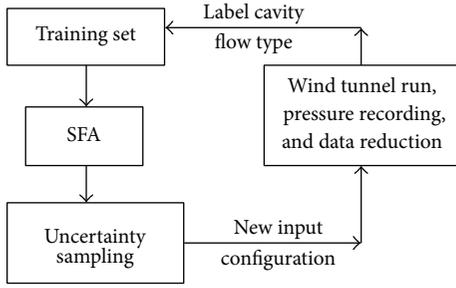
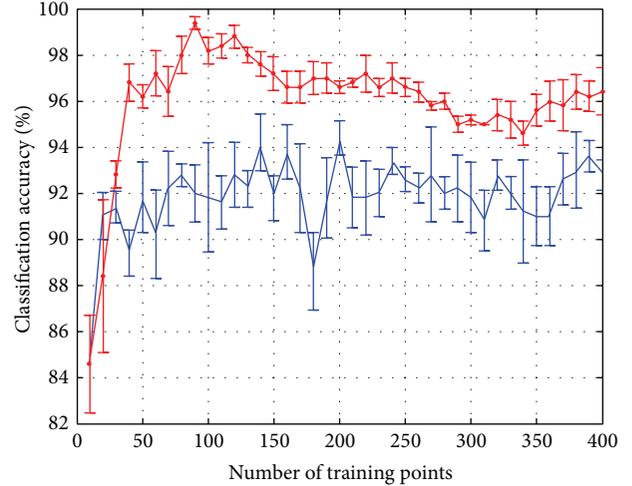


FIGURE 6: Schematic representation of active learning.

where N_{test} = number of points in the test set and m_{test} = number of misclassifications. Results shown in Figure 7 were obtained by using SFA in conjunction with uncertainty sampling to select new samples for labeling. The red and the blue curves in Figure 7(a) show the increasing percentage prediction accuracy on a fixed test set (100 points) as training points were incremented sequentially using active learning and random sampling, respectively. Initial approximations for both sampling strategies were constructed using ten points chosen by LHS design and two points were chosen at a time for labeling from a pool of 400 regularly spaced grid points. This process was repeated 50 times to eliminate any bias due to the choice of the initial set of training points. Figure 7(a) shows mean percentage prediction accuracy computed over 50 permutations. Error bars show the standard error. Figure 7(b) shows the location of the data points chosen by the active learning scheme in one of the permutations. The active learning scheme clearly performs better than random sampling. Since we have an unlimited supply of unlabeled data points, the learning curves for active and passive learning do not converge.

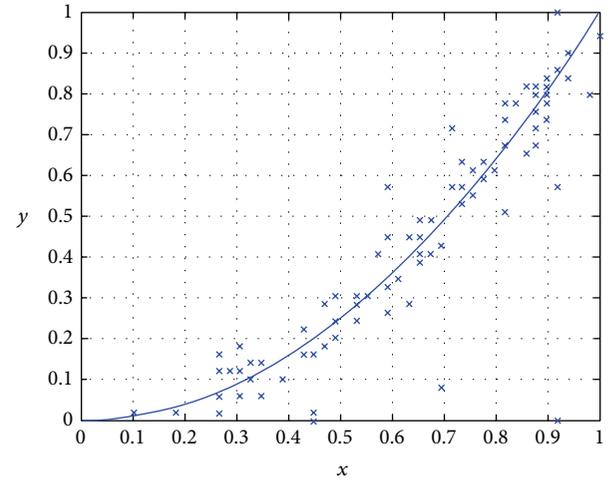
As previously mentioned, multiclass classification problems are commonly attempted by combining several binary classifiers with a *one versus all* or *one versus one* approach. Selecting data points lying close to the classification boundaries in these problems would not be an optimal approach because one point could be informative for two classes but uninformative for the rest. A simple way to avoid this problem is to use the posterior probability estimates of the binary classifiers to pick a sample for labeling. According to uncertainty sampling, an informative sample would be one that has the lowest classification uncertainty. Several authors have suggested picking a sample for labeling that bears the minimum difference between the highest and the second highest posterior probability estimates [27, 28]. In this work, this method is used to select the next sample for labeling.

SFA is a deterministic classifier that attempts to directly estimate the discriminant function of a binary classifier. Like Support Vector Machines (SVMs), SFA does not output posterior probabilities of a test point belonging to a particular class. Platt [29] introduced a method to directly train the parameters of a sigmoid function to map the deterministic SVM outputs into posterior probabilities. Several authors



— Passive
— Active

(a)



(b)

FIGURE 7: (a) Mean percentage classification accuracy versus number of training points. Error bars show standard error. (b) Crosses represent the training points chosen by the active learning scheme, and solid line is the class discrimination boundary $y = x^2$.

extended this notion to a softmax function for multiclass classification problems [30] given by

$$P(Y = k | \xi_i) = P_k^i = \frac{\exp [a_k u_n^{ak}(\xi_i) + b_{0,k}]}{\sum_{j=1}^M \exp [a_j u_n^{aj}(\xi_i) + b_{0,j}]} \quad (14)$$

Here, u_n^{ak} is the real valued output of the k th binary classifier at ξ_i , and a_k and $b_{0,k}$ are parameters of the softmax function. The parameters of the softmax function are determined by maximizing the following log-likelihood function given in

$$\max_{a_k, b_{0,k}} \sum_{i=1}^s \log [P_k^i] \quad (15)$$

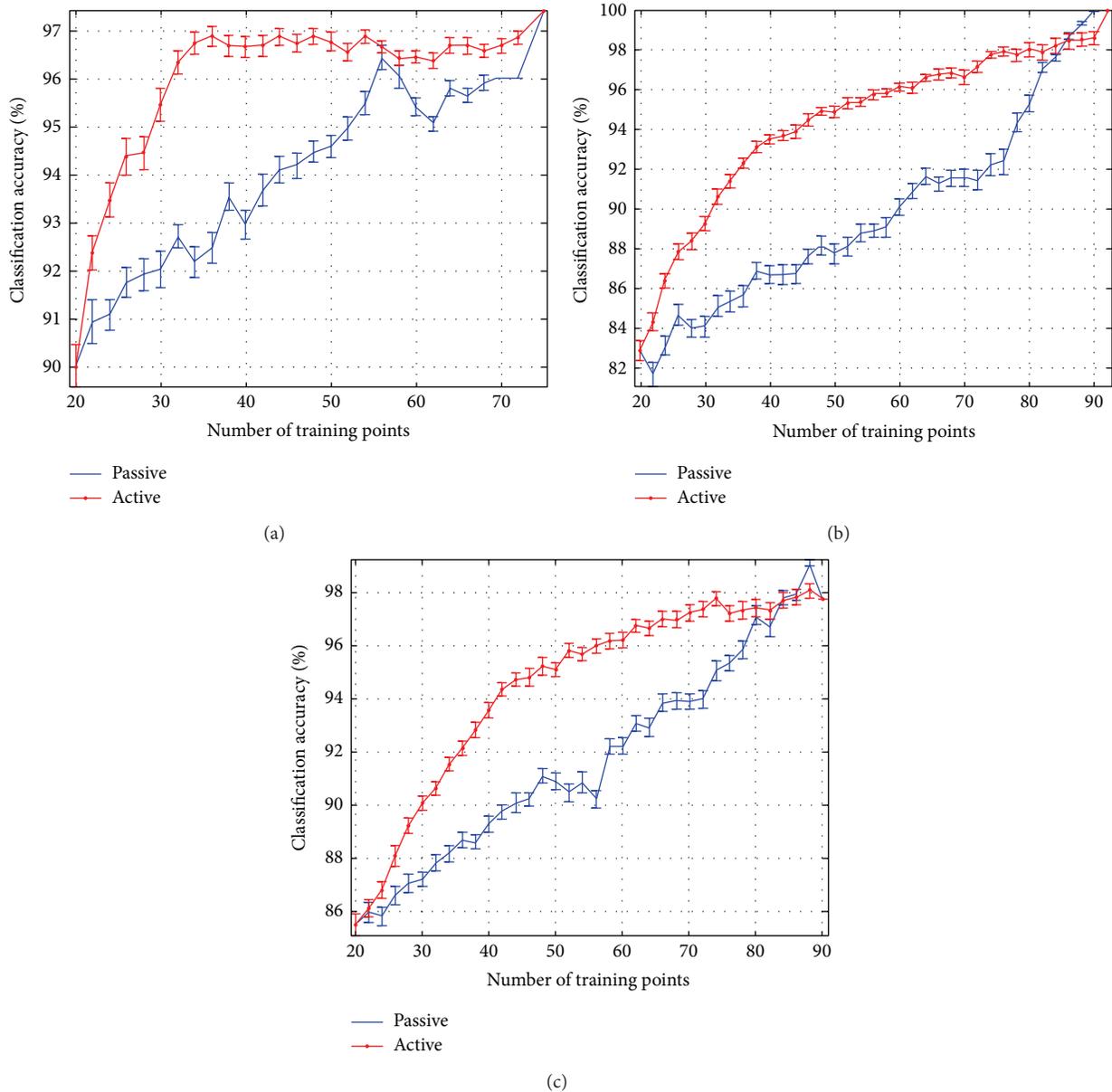


FIGURE 8: Mean cavity flow type prediction accuracy of active and passive learning at (a) $w/h = 1$, (b) $w/h = 4$, and (c) $w/h = 8$. Error bars show the standard error.

3. Results

A total of 267 wind tunnel runs were conducted by Tracy and Plentovich [1] with the resulting data plotted in Figure 5. Percentage errors were computed in the same manner as that presented in the previous section. The error convergence graphs shown in Figure 8 were obtained by increasing the number of training points and testing the constructed networks on all 267 test points. We compared the misclassification error rate of SFA with training points chosen by US against training points chosen by LHS design. The purpose of this comparison was to demonstrate that the active machine learning algorithm achieves better generalization ability, that is, better cavity flow type prediction accuracy with fewer

training points. We treat the current problem as a 3-class classification problem and focus on generating optimal M_∞ and l/h combinations for each w/h . We do not focus on determining optimal w/h ratio because of its insufficient resolution in the available data.

The active machine learning algorithm initially needs a few training points to construct a hypersurface to determine the most uncertain data points. There is no principled approach to determine the optimum number of initial points, so we used 20 initial points to start the active learning procedure. In addition, a heuristic approach to choose an initial number of training points is validated because by construction the active learning scheme adds points that correspond to maximum information gain which leads to low

TABLE I: Comparison of classification accuracies.

w/h	Number of training points	LHS (%)	US (%)
1	38	94 ± 0.3	97 ± 0.2
4	46	88 ± 0.4	95 ± 0.3
8	44	90 ± 0.4	95 ± 0.3

errors of the metamodel only with a few carefully selected points. In other words, the initial number of training points has no effect on the rate of convergence of the active learning scheme. These points were chosen by the LHS method on two dimensions M_∞ and l/h . To eliminate any biases due to the randomness in the LHS designs, we repeated the training and testing 50 times. Two points were added to the training set each time and the active data selection process continued until all of the available points were used. This approach allows the user to choose a single point or a batch of points per iteration. We decided to add two points per iteration to save computational time spent to complete 50 random permutations used to eliminate bias in the results.

Figure 8 shows that active learning clearly outperforms the passive LHS technique at all three w/h ratios. Active learning accuracy increases sharply and then gradually converges with LHS classification accuracy as unlabeled samples are exhausted. With 40 training data points at $w/h = 1$, active learning has a classification accuracy of $97 \pm 0.2\%$, while LHS has $93 \pm 0.3\%$. This means that if the user decides to conduct 40 wind tunnel runs at $w/h = 1$, then actively sampling input configurations by the US technique would result in data that has more information than data sampled passively by the LHS technique. Training data sampled from critical regions of the input-output hyperspace give more generalization ability to SFA than the training data sampled just from the input space.

The only disadvantage US based active learning has compared to passive LHS technique is its greater computational expense. In this problem, the active learning scheme took only a few minutes longer compared to the passive learning scheme. However, the relatively high cost of wind tunnel testing more than justifies the increased computational cost. The active learning curve for $w/h = 1$ flattens out at about 35 training points because the US algorithm chooses almost all of the points lying close to the classification boundaries in the first 8 iterations. Table 1 shows cavity flow type classification accuracy of SFA, using half of the data points for training, sampled by the US and LHS techniques.

4. Conclusions

This paper has demonstrated that active machine learning tools can be used to design wind tunnel runs to measure unsteady pressure measurements in a cavity flow classification problem. We propose that a machine learning tool be used in conjunction with wind tunnel testing to guide the data collection procedure. We believe that a mathematical tool that samples data using the input-output relationship should be more effective than a tool that just samples data from the input domain. Such a technique can save wind tunnel testing time and cost at relatively little computational

expense. In particular, we demonstrated the use of SFA with the uncertainty sampling technique on a multiclass cavity flow type prediction problem. Results were compared against passively collected data using the traditional LHS technique. We believe that active machine learning tools have the potential to help engineers to accelerate through wind tunnel testing by steering through the test matrix in an incremental and optimal manner.

Nomenclature

a, b_0 :	Parameters of the softmax function
b_n :	n th bias in the approximation
C_p :	Pressure coefficient
M :	Total number of classes
c_n :	n th linear coefficient in the approximation
d :	Input dimension
E :	Expectation operator
f :	Arbitrary function
$F[v]$:	Discriminant function; proxy output of v
j :	Component index of $\mathbf{r}_{(n-1)}$ with the maximum absolute magnitude
k :	Associated with the number of classes
l/h :	Length over depth ratio
M_∞ :	Freestream Mach number
n :	Number of bases
P :	Posterior probability
r_n :	n th stage of the function residual
\mathbb{R} :	Real coordinate space
\mathbb{R}^d :	d -dimensional vector space
s :	Number of training samples
$\text{sign}[v]$:	Sign function, +1 if $v > 0$ and -1 if $v < 0$
$u(\xi)$:	Target function
$u_n^a(\xi)$:	n th stage of the target function approximation
$u_n^{ak}(\xi)$:	n th stage approximation from the k th classifier
v :	Dummy variable
w/h :	Width over depth ratio
Y :	Output class
$\{f, \dots\}$:	Vector of the elements of f
$\langle f \rangle$:	Sum of the elements of f
$\langle f, v \rangle_D$:	$\sum_i^s (f_i v_i) =$ discrete inner product of f and v
$ f $:	Absolute value of f
$\ f\ _2$:	$(\int_\Omega f^2 d\xi)^{1/2} = L_2$ norm of f
$\ f\ _{2,D}$:	$(\sum_i^s (f_i)^2)^{1/2} =$ discrete L_2 norm of f .

Greek Symbols

α :	Width parameter of Gaussian basis function
β :	Set of nonlinear optimization parameters
ε :	Objective function
θ :	Vector of parameters of an arbitrary data fitting model
ξ :	d -dimensional input of the target function
ξ_i :	i th sample input of the target function
ξ_n^* :	Sample input with component index j^* at the n th stage

σ : Width parameter of Gaussian radial basis function
 τ : User specified tolerance
 ϕ : Basis function.

Subscripts

i : Dummy index
 j^* : Component index of $\mathbf{r}_{(n-1)}$ with the maximum magnitude
 n : Associated with the number of bases
 s : Associated with the number of samples.

Superscripts

a : Associated with the approximated function
 d : Associated with the input dimension.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Support for this work was provided by the NASA Ames Research Grant NCC-2-8077 and NASA Cooperative Agreement no. NCC-1-02038.

References

- [1] M. B. Tracy and E. B. Plentovich, "Cavity unsteady-pressure measurements at subsonic and transonic speeds," Tech. Rep. NASA-TP-3669, 1997.
- [2] <http://futureweaponsandmore.blogspot.com/>.
- [3] <http://wararchive.blogspot.com/2009/11/lockheed-f-35-lightning-ii-specs-and.html>.
- [4] K.-M. Chung, "Geometric effect on compressible rectangular cavity flows," *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 45, no. 147, pp. 28–34, 2002.
- [5] N. M. Komerath, K. K. Ahuja, and F. W. Chambers, "Prediction and measurement of flows over cavities—a survey," in *Proceedings of the AIAA 25th Aerospace Sciences Meeting*, AIAA 87-0166, Reno, Nev, USA, January 1987.
- [6] A. Srivastava and A. J. Meade, "Application of sequential function approximation based machine learning for classification of cavity flow and resonance," *Journal of Aircraft*. In press.
- [7] A. Srivastava, A. J. Meade, and J. Needham, "Adaptive algorithm for classifying launch and recovery of the HH-60H seahawk," *Journal of Aerospace Computing, Information and Communication*, vol. 6, no. 2, pp. 109–120, 2009.
- [8] M. Kokkolaras, *Utilizing parallel optimization in computational fluid dynamics [Ph.D. thesis]*, Mechanical Engineering Department, Rice University, 1997.
- [9] A. J. Meade Jr., "Application of scattered data approximation to a rotorcraft health monitoring problem," *Journal of Aerospace Computing, Information and Communication*, vol. 1, no. 4, pp. 175–188, 2004.
- [10] A. Srivastava and A. J. Meade, "A sparse greedy algorithm for classification," *Advances in Adaptive Data Analysis*, vol. 2, no. 1, pp. 97–114, 2010.
- [11] M. Hasenjäger, *Active Data Selection in Supervised and Unsupervised Learning*, Der Technischen Fakultät, der Universität Bielefeld, 2000.
- [12] D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3–12, 1994.
- [13] M. D. McKay, "Latin Hypercube Sampling as a tool in uncertainty analysis of computer models," in *Proceedings of the Winter Simulation Conference*, pp. 557–564, 1992.
- [14] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics, 2001.
- [15] D. A. Cohn, "Neural network exploration using optimal experiment design," *Neural Networks*, vol. 9, no. 6, pp. 1071–1083, 1996.
- [16] Y. Freund, "Boosting a weak learning algorithm by majority," *Information and Computation*, vol. 121, no. 2, pp. 256–285, 1995.
- [17] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [18] L. K. Jones, "Constructive approximations for neural networks by sigmoidal functions," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1586–1589, 1990.
- [19] L. K. Jones, "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training," *The Annals of Statistics*, vol. 20, no. 1, pp. 608–613, 1992.
- [20] A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 930–945, 1993.
- [21] C. A. J. Fletcher, *Computational Galerkin Methods*, Springer, New York, NY, USA, 1984.
- [22] A. J. Meade Jr. and B. A. Zeldin, "Approximation properties of local bases assembled from neural network transfer functions," *Mathematical and Computer Modelling*, vol. 28, no. 9, pp. 43–62, 1998.
- [23] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.
- [24] B. Cui, H. Lin, and Z. Yang, "Uncertainty sampling-based active learning for protein-protein interaction extraction from biomedical literature," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10344–10350, 2009.
- [25] G. Schohn and D. Cohn, "Less is more: active learning with support vector machines," in *Proceedings of 17th International Conference on Machine Learning*, pp. 839–846, 2000.
- [26] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 287–294, Pittsburgh, Pa, USA, July 1992.
- [27] T. Luo, K. Kramer, D. B. Goldgof et al., "Active learning to recognize multiple types of plankton," *Journal of Machine Learning Research*, vol. 6, pp. 589–613, 2005.

- [28] C. Korner and S. Wrobel, "Multi-class ensemble-based active learning," *Machine Learning*, vol. 42, no. 12, pp. 687–694, 2006.
- [29] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods," in *Advances in Large Margin Classifiers*, pp. 61–74, MIT Press, 1999.
- [30] K. Duan, S. S. Keerthi, W. Chu, S. K. Shevade, and A. N. Poo, "Multi-category classification by soft-max combination of binary classifiers," in *Proceedings of the 4th International Workshop on Multiple Classifier Systems*, pp. 125–134, 2003.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

