Struct Multidisc Optim (2013) 47:479–491
DOI 10.1007/s00158-012-0857-6

RESEARCH PAPER

Adaptive virtual support vector machine for reliability analysis of high-dimensional problems

Hyeongjin Song · K. K. Choi · Ikjin Lee ·
Liang Zhao · David Lamb

Received: 21 December 2011 / Revised: 26 October 2012 / Accepted: 29 October 2012 / Published online: 28 November 2012

Abstract In this paper, an efficient classification methodology is developed for reliability analysis while maintaining an accuracy level similar to or better than existing response surface methods. The sampling-based reliability analysis requires only the classification information—a success or a failure—but the response surface methods provide function values on the domain as their output, which requires more computational effort. The problem is even more challenging when dealing with high-dimensional problems due to the curse of dimensionality. In the newly proposed virtual support vector machine (VSVM), virtual samples are generated near the limit state function by using an approximation method. The function values are used for approximations of virtual samples to improve accuracy of the resulting VSVM decision function. By introducing the virtual samples, VSVM can overcome the deficiency in existing classification methods where only classification values are used as their input. The universal Kriging method is used to obtain virtual samples to improve the accuracy of the decision function for highly nonlinear problems. A sequential sampling strategy that chooses new samples near the limit state function is integrated with VSVM to improve the accuracy. Examples show the proposed adaptive VSVM yields better efficiency in terms of modeling and response evaluation time and the number of required samples while maintaining similar level or better accuracy, especially for high-dimensional problems.

Keywords Surrogate model · Support vector machine (SVM) · Sequential sampling · Virtual samples · Virtual support vector machine (VSVM) · High-dimensional problem

1 Introduction

Accurate reliability analysis is of great importance for solving engineering problems. Inaccurate reliability analysis result can lead to an unreliable or overly conservative design. Numerous methods that are based on the most probable point (MPP) are available in literature for carrying out reliability analyses of many engineering problems, for which the sensitivity information can be obtained (Haldar and Mahadevan 2000; Tu et al. 1999; Youn et al. 2005). On the other hand, the sensitivity is often not available or difficult to obtain accurately in complex multi-physics or multidisciplinary design problems. Without sensitivity, an alternative to the MPP-based reliability analysis method is to directly perform the probability integration numerically by carrying out computer simulations at the Monte

H. Song · K. K. Choi (✉)
Department of Mechanical & Industrial Engineering,
College of Engineering, The University of Iowa,
Iowa City, IA 52242, USA
e-mail: kkchoi@engineering.uiowa.edu

H. Song
e-mail: hyesong@engineering.uiowa.edu

I. Lee
Department of Mechanical Engineering,
The University of Connecticut, Storrs, CT 06269, USA
e-mail: ilee@engr.uconn.edu

L. Zhao
Schlumberger, 1310 Rankin Rd, Houston, TX 77073, USA
e-mail: lzhao01@slb.com

D. Lamb
US Army RDECOM/TARDEC, Warren, MI 48397–5000, USA
e-mail: david.lamb@us.army.mil

Carlo simulation (MCS) sampling points (Rubinstein 1981; Ching 2011). However, this method requires a large number of response function evaluations and can be impractical in terms of computational cost.

Therefore, surrogate-based methods are used to reduce the computational cost without requiring sensitivity analysis. The main advantage of the surrogate-based method is that a limited number of function evaluations can be used to construct surrogate models. Many different surrogates, such as polynomial response surface (PRS), radial basis function (RBF), multivariate adaptive regression spline (MARS), moving least squares (MLS) and Kriging, have been developed and applied to engineering problems (Cressie 1991; Barton 1994; Jin et al. 2001; Simpson et al. 2001; Wang and Shan 2007; Forrester et al. 2008; Forrester and Keane 2009; Zhao et al. 2011). These surrogates provide approximations of otherwise expensive computer simulations. Once an accurate surrogate model is generated, MCS can be carried out using the surrogate model to estimate the reliability. This method is called the sampling-based reliability analysis or sampling-based reliability-based design optimization (RBDO). In the sampling-based RBDO, probabilistic constraints are used. Thus, not only the probability of failure but also its sensitivity needs to be accurately estimated.

In Zhao et al. (2011), the dynamic Kriging method was developed and applied successfully for sampling-based methods. However, when the response values are evaluated, all samples within the design space are used to calculate the trend and random component of the Kriging method. Therefore, response evaluations using the Kriging method could be computationally expensive since the sampling-based reliability analysis and RBDO need response evaluations at a very large number of MCS points to accurately estimate the probability of failure and its sensitivity. Furthermore, surrogate-based approaches usually obtain response function values over the entire domain. Therefore, the surrogate-based method requires a large number of samples even at the unnecessary regions to reach the target accuracy (i.e., mean squared error or R^2), and thus they become inefficient (Hurtado and Alvarez 2003). The computational burden becomes heavier in high-dimensional space due to the curse of dimensionality (Vapnik 1998; Cherkassky and Mulier 1998; Burges 1998). Therefore, a classification method with simpler formulation needs to be investigated while achieving similar accuracy. There exist several sequential sampling methods that select new samples near the constraint boundary using Gaussian process models (Bichon et al. 2008; Bichon et al. 2011; Ranjan et al. 2008; Lee and Jung 2008; Bect et al. 2012; Viana et al. 2012).

The support vector machine (SVM) is a classification method, so it constructs only the decision (i.e., limit state) function, which maximizes the distance to the existing samples (Vapnik 1998, 2000; Cherkassky and Mulier 1998; Scholkopf 1999; Kecman 2001; Scholkopf and Smola 2002). In SVM, only support vectors are used instead of all samples for evaluations of responses, with the most calculations performed through the kernel function. Thus, the response evaluation process is very efficient for MCS, compared to the Kriging method. Another advantage of the classification method is that they can deal with multiple constraints at once (Basudhar et al. 2012). The SVM with a sequential sampling strategy, which is called the explicit design space decomposition (EDSD), is developed and applied to discontinuous and disjoint problems successfully (Basudhar et al. 2012; Basudhar and Missoum 2008, 2010). Even though EDSD can be also used for continuous and differentiable problems, it often converges very slowly, and thus requires a large number of samples. One of the main reasons for the inefficiency of EDSD for continuous problems is that it only uses the classification response function values rather than the function values to construct the decision function. Therefore, the accuracy needs to be improved for continuous problems.

Accurate probability of failure can be obtained from accurate limit state function, and accurate sensitivity can be calculated by using the score function (Lee et al. 2011). It is interesting to note that the score function depends on the derivatives of the input joint and marginal distributions. Therefore, the sampling-based method requires only an accurate decision function to evaluate the probability of failure and its sensitivity. That is, only the decision between a success and a failure is used instead of the function value. Thus, even though SVM, being a classification method, cannot be directly used for deterministic design optimization due to lack of design sensitivity, it is applicable for sampling-based RBDO.

In this paper, a virtual SVM (VSVM) is proposed to improve the accuracy of SVM, while maintaining the desirable features of SVM, by using the available response function values. Unlike EDSD, VSVM is developed primarily for continuous problems. The VSVM constructs the decision function rather than the surrogate model over the given domain. The proposed adaptive sampling method provides new samples near the limit state, which makes the method efficient. The proposed method provides an explicit form of the limit state function, so it is efficient in obtaining response values at MCS points. Basic concepts and important features of SVM are presented in Section 2. In Section 3, the virtual sample generation method and the adaptive sampling strategy are explained. Stopping criteria are defined for the updating process as the target accuracy is achieved. In Section 4, EDSD and the dynamic Kriging methods are compared with the proposed VSVM to demonstrate the efficiency of VSVM while maintaining the accuracy. An error measure is also defined to compare the accuracy of the result. The conclusion follows in Section 5.

2 Support vector machine

An SVM is a machine-learning concept used for the classification of data in pattern recognition (Vapnik 1998, 2000; Cherkassky and Mulier 1998; Scholkopf 1999; Kecman 2001; Scholkopf and Smola 2002; Basudhar et al. 2012; Basudhar and Missoum 2008, 2010). It has the ability to explicitly construct a multidimensional and complex decision function that optimally separates multiple classes of data. Even though SVM is able to deal with multi-class cases, only two classes—success or failure—are used in reliability analyses, and thus only a two-class classification problem will be considered in this paper. Good features for the high-dimensional problem make SVM an appropriate method for the formulation of the explicit limit state function. In this section, a brief overview of SVM is presented, including basic ideas and some important features.

2.1 Linear SVM

For the given multidimensional problem, N samples are distributed within the given window. Each sample $\mathbf{x}_i$ is associated with one of two classes characterized by values $y_i = \pm 1$, which represents a success (-1) or a failure $(+1)$. The SVM algorithm constructs the decision function that optimally separates two classes of samples. The corresponding linear explicit boundary function is expressed as

$$s(\mathbf{x}) = b + \sum_{i=1}^{N} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) \tag{1}$$

where b is the bias, α_i are Lagrange multipliers obtained from the quadratic programming optimization problem used to construct SVM, and $\mathbf{x}$ is an arbitrary point to be predicted. The classification of any arbitrary point $\mathbf{x}$ is given by the sign of $s(\mathbf{x})$ in (1). The optimization process is used to solve for the optimal SVM decision function with a maximal margin. Figure 1 shows a linear SVM result, and the notion of margin can be easily noticed. In this case, the margin is the shortest distance between two parallel hyperplanes given by $s(\mathbf{x}) = \pm 1$ in the design space. These hyperplanes are called support hyperplanes and pass through one or several samples, which are called support vectors. The SVM optimization process also does not allow any samples to exist between two hyperplanes.

The Lagrange multipliers associated with the support vectors are positive while the other Lagrange multipliers are zero. It means that the explicit SVM decision function uses only support vectors in its formulation, and thus an SVM constructed only with support vectors is identical to the one obtained with all samples. Typically, the number of support vectors is much smaller than the number of samples N.

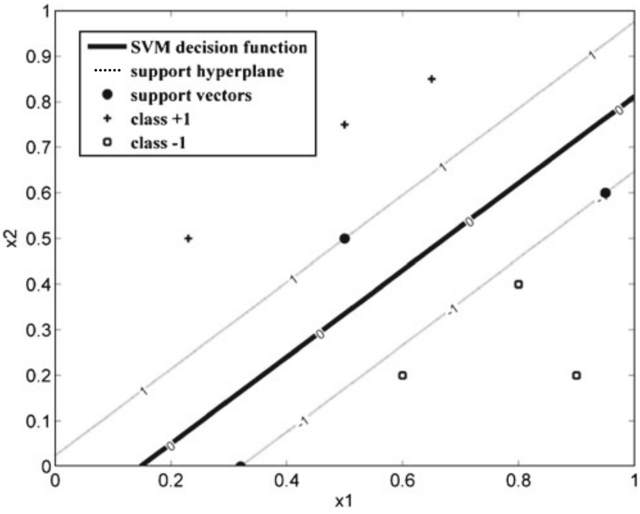


Fig. 1 Linear decision function for two-dimensional problem

2.2 Nonlinear SVM and kernel functions

To construct nonlinear decision functions, kernels are introduced in SVM. In the formulation of the SVM decision function, it is assumed that there exists a higher-dimensional space where the transformed data can be linearly separated. The transformation from the original design space to the higher-dimensional space is based on the kernel function $K(\mathbf{x}_i, \mathbf{x})$ in SVM. The nonlinear decision function is expressed as

$$s(\mathbf{x}) = b + \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) \tag{2}$$

Instead of the linear function $(\mathbf{x}_i \cdot \mathbf{x})$ in (1), the nonlinear $K(\mathbf{x}_i, \mathbf{x})$ is used in the form of nonlinear decision function. Therefore, $s(\mathbf{x})$ is linear with respect to $K(\mathbf{x}_i, \mathbf{x})$ and nonlinear with respect to $\mathbf{x}$.

The kernel $K(\mathbf{x}_i, \mathbf{x})$ in the SVM equation can have different forms, such as polynomial, Gaussian, Sigmoid, etc. The Gaussian kernel is known to have fewer numerical difficulties than other kernels, so it is used in this paper (Hsu et al. 2004). The form of the Gaussian kernel is given as (Vapnik 1998, 2000; Cherkassky and Mulier 1998; Scholkopf 1999; Kecman 2001; Scholkopf and Smola 2002):

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2}\right) \tag{3}$$

where σ is the parameter of the Gaussian kernel. Figure 2 is an example of nonlinear SVM decision function with the Gaussian kernel for a two-dimensional problem. Even though the boundary is always linear in the transformed

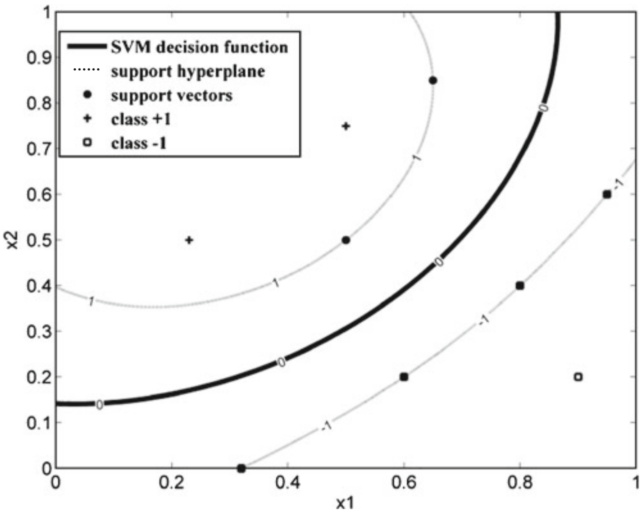


Fig. 2 Nonlinear decision function for two-dimensional problem

higher-dimensional space, the boundary is nonlinear in the original design space. In this paper, *SVM and Kernel Methods Matlab toolbox* (Canu et al. 2005) is used for the formulation of SVM.

The SVM can deal with high-dimensional problems and can separate two classes of data with the maximal margin. The SVM decision function has an explicit form, and thus predictions based on SVM are faster for MCS than those based on implicit surrogate methods such as the Kriging method. The prediction speed is important for sampling-based RBDO, since a very large number of MCS points are required in evaluating the probability of failure.

3 Virtual support vector machine

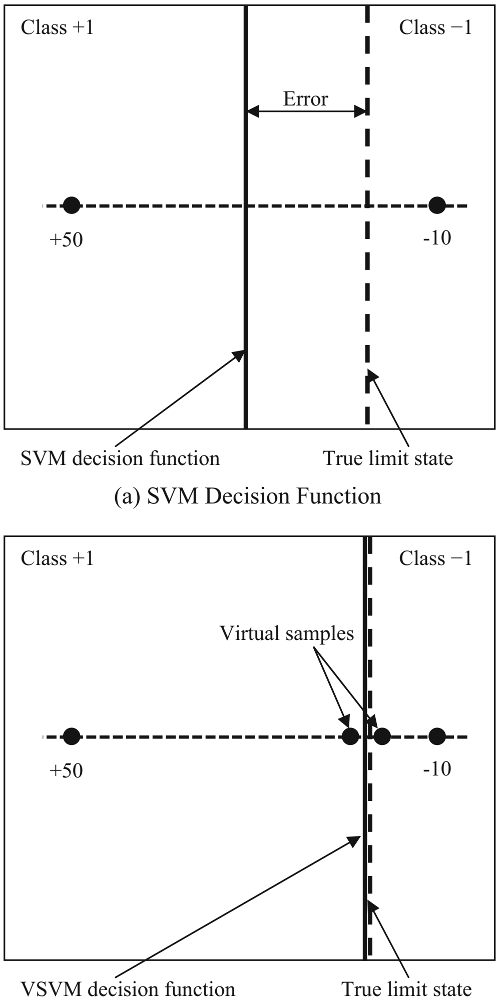
The EDSD, which is based on the conventional SVM, yields good performance for discontinuous limit state functions. However, there exists a limitation for continuous problems. Since EDSD does not use function values, EDSD converges very slowly, and consequently requires many samples in dealing with continuous problems. Therefore, the accuracy needs to be improved for continuous problems, which can be achieved by inserting virtual samples generated based on available function values.

3.1 Virtual sample generation and VSVM

For the construction of SVM, initial samples, which include both success and failure samples, should be given. Initial samples are generated by Latinized Centroidal Voronoi Tessellation (LCVT), since it shows very good uniformity and randomness (Basudhar et al. 2012; Saka et al. 2007).

Since SVM deals only with classification of responses, i.e., successes (-1) or failures ($+1$), the SVM decision function is usually located in the middle of opposite signed samples regardless of the function values of the given samples as shown in Fig. 3a. However, in reality, samples with small absolute function values are more likely to be located closer to the limit state function than those with large absolute function values.

In this paper, two types of samples are used. The first type is real samples, which include initial samples and sequential samples. Sequential samples are inserted when the VSVM model does not meet the accuracy requirement. These real samples require function evaluations. The second type is virtual samples, which are generated using an approximation method to improve the accuracy of the VSVM decision function. Such virtual samples do not require real function evaluations and only have virtual signs. The basic idea of



(a) SVM Decision Function

(b) VSVM Decision Function with Virtual Samples

Fig. 3 SVM decision function and VSVM decision function with virtual samples—*solid lines*

VSVM is to increase the probability of locating the decision function close to the limit state function, by inserting two opposite signed virtual samples between the given two samples. These virtual samples play two major roles in VSVM. One is to make the predictions more accurate, and the other is to locate new sequential samples near the limit state function, which will be presented in Section 3.2. In Fig. 3b, the VSVM decision function is shifted towards the sample with a small absolute function value by inserting two virtual samples. The virtual samples with opposite signs should be near the limit state function and be equally distanced from the limit state function to obtain the best VSVM decision function.

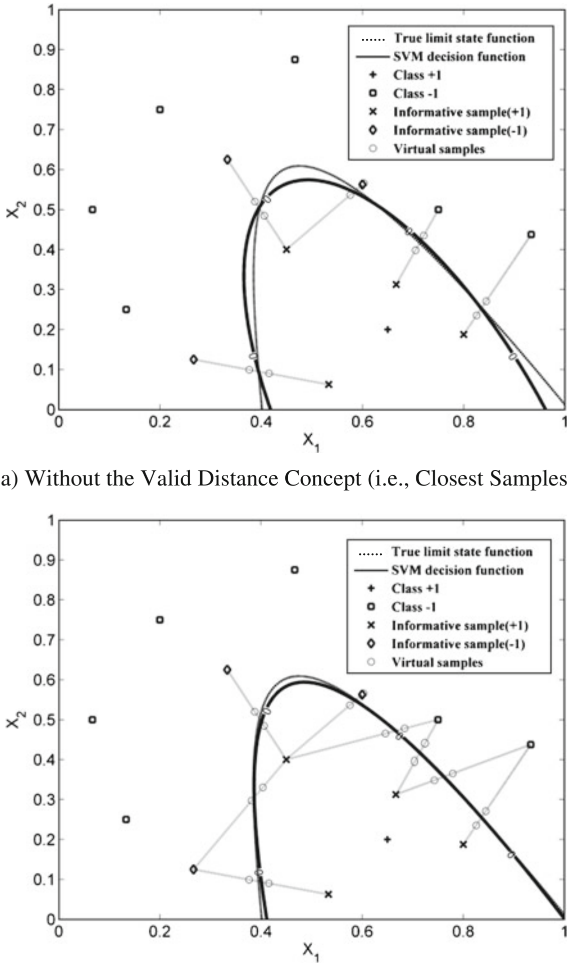
3.1.1 Informative sample set and valid distance

Virtual samples may be generated from the approximation method using any pair of real samples. However, it is desirable to use two opposite class samples. If both samples have the same sign, then finding the decision function requires an extrapolation, which is often inaccurate and the decision function is not located between two given samples. If two existing samples have opposite signs (+1 and −1), then the decision function should exist between the two samples for the continuous problem. Even though any pair of different class samples can be used in theory, if the distance between two given samples is large or both samples are far from the limit state function, then accurate positioning of the zero point between two samples cannot be expected. Thus, at least one of two points should be close to the limit state function, and both should be close to each other to make approximations more accurate and useful. Therefore, an informative sample set, from which virtual samples are generated, is defined first. Original SVM is constructed first based on existing samples to identify support vectors. Support vectors are located near the limit state function, and thus they are included in the informative set. It is highly probable that some samples with small absolute values are also located close to the limit state function, even though they may not be support vectors. Therefore, all samples that have absolute response values that are smaller than the maximum absolute responses of the support vectors are chosen as members of the informative set. The informative sample set can be expressed as

$$\begin{aligned} &f_{\max} = \max_i \left(\left|f\left(\mathbf{x}_i^*\right)\right|\right), \ i = 1, \cdots, N_{\mathrm{SV}} \\ &\left\{\mathbf{x}_j \,\middle|\, \left|f\left(\mathbf{x}_j\right)\right| \le f_{\max}, \ j = 1, \cdots, N\right\} \end{aligned} \tag{4}$$

where $\mathbf{x}_i^*$ is the ith support vector, N_{sv} is the number of support vectors, $\mathbf{x}_j$ is the jth sample, N is the number of samples, and f is the function value at the given sample.

From the previously chosen informative samples, the closest opposite signed samples are paired to generate virtual samples between each pair. However, there exist some pairs that can generate important virtual samples, even though they do not belong to the closest pairs. To solve this problem, a valid distance concept is introduced. Pairs can generate virtual samples if the distance between them is shorter than the valid distance. If the valid distance is too large, then there is a risk of including many unnecessary virtual samples and producing poor approximations. If the valid distance is too short, it may not include information that is more useful. To introduce more informative virtual samples while maintaining virtual samples from the closest pairs, the largest distance among all distances between the closest pairs is defined as the valid distance. Figure 4 shows the influence of the valid distance concept in a two-dimensional example. By inserting three additional pairs of virtual samples, the accuracy is improved in the area near the new virtual sample pairs.



(a) Without the Valid Distance Concept (i.e., Closest Samples Only)

(b) With the Valid Distance Concept

Fig. 4 VSVM decision functions without/with valid distance concept

3.1.2 Approximations for zero positions

Two additional steps are needed for the generation of the virtual samples after the informative sample set and the valid distance are defined. Firstly, since the true limit state function is not known in general, a zero position is approximated from two different class samples by using approximation methods such as linear approximation, Kriging, MLS, etc. A zero position means a point where the approximation value is zero among all the points on the line between two opposite signed samples. A linear approximation simply assumes that the function value between two given samples is linear and finds the zero point. The linear approximation is very efficient and easy to apply but can be inaccurate for highly nonlinear functions.

The Kriging or MLS methods are accurate near existing samples, so they are appropriate to obtain better approximations, especially for highly nonlinear functions. In this paper, the universal Kriging method is used by using all existing samples to approximate zero points between two opposite signed samples, and the SURROGATES toolbox (Viana 2010) is used for the construction of the universal Kriging model. The optimization problem for finding the zero position between two samples is expressed as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \left|\widehat{f}(\mathbf{x})\right| \\ \text{s.t.} \quad & \mathbf{x} = \mathbf{x}_i \cdot t + \mathbf{x}_j \cdot (1-t) \\ & 0 \le t \le 1 \end{aligned} \tag{5}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are original samples with opposite signs, $\mathbf{x}$ is a point on the straight-line connecting $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\widehat{f}(\mathbf{x})$ is an approximated value at $\mathbf{x}$ obtained by the universal Kriging method.

It requires a fair amount of computational time to solve (5) accurately. In particular, Kriging approximations take a large amount of time if approximations are calculated one by one due to its implicit formulation. Therefore, the line connecting two opposite signed samples $\mathbf{x}_i$ and $\mathbf{x}_j$ is divided into multiple elements, their Kriging approximations are evaluated all at once, and the position with the minimum absolute function value is chosen. The size of one element needs to be smaller than the virtual margin, which is the distance between a pair of virtual samples, to generate an accurate surrogate. Thus, the number of elements depends on the virtual margin. Then vector calculation can be carried out all at once rather than one-by-one calculation, which is more efficient in Matlab. In this way, the elapsed time for generating virtual samples is reduced from 39.94 s to 2.01 s per iteration for a twelve-dimensional problem.

In the Kriging model, the correlation function $\mathbf{R}(\boldsymbol{\theta}, \mathbf{x}_i, \mathbf{x}_j)$ should be estimated from the sample data, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are given samples and $\boldsymbol{\theta}$ is the process parameter. The influence of the parameter $\boldsymbol{\theta}$ on the performance is significant, and thus the determination of the parameter is important. To find the optimum $\boldsymbol{\theta}$, different methods such as the Hookes&Jeeves (H-J), Lavenberg-Marquardt (L-M), genetic algorithm (GA), and Pattern Search (PS) methods (Viana 2010; Martin 2009; Lewis and Torczon 1999) have been applied. Among them, the PS method is most accurate, but it requires more computational effort than other methods. However, with VSVM, a smaller number of iterations can be used to achieve a similar level of accuracy with more accurate Kriging models by locating new samples correctly. Therefore, time and resources can be saved by using the PS method.

To make the estimation process more efficient, the history of parameter changes was investigated to find that the new optimum $\boldsymbol{\theta}$ is close to the previous optimum $\boldsymbol{\theta}$ in general. If the current VSVM model is similar to the previous VSVM, then both optimum Kriging parameters are also close to each other. Therefore, the previous optimum Kriging parameter $\boldsymbol{\theta}$ value is used as the initial value for the PS method. By implementing this efficiency strategy, the elapsed time to find the optimum $\boldsymbol{\theta}$ is reduced by 90 % per iteration on average.

3.1.3 Generation of virtual samples from zero positions

Secondly, two opposite signed virtual samples are generated near the zero point. One is located in the direction of the success sample, and the other is in the direction of the failure sample. Both virtual samples should be between the given two opposite signed samples and on the line that connects these points, as shown in Fig. 3. Then, a new SVM decision function based on the original and virtual samples will be located between the virtual sample pairs, because the virtual samples in each pair have different signs and are close to each other. If approximations for zero points are accurate, then both virtual samples and a new decision function will be near the limit state function.

One important question is how closely a pair of virtual samples should be located. If the distance between a pair of virtual samples is too long, then these virtual samples will not be chosen as support vectors and they become ineffective. To make the virtual samples effective, the distance should be short enough that the virtual samples are selected as support vectors so that the supporting hyperplanes are located near the decision function. If a smaller target error is required, the virtual margin needs to be reduced accordingly.

If many virtual samples are clustered together within a small region, the additional information from the most closely located virtual samples is negligible, and the computational time increases. In selection of virtual samples, the trade-off between the amount of additional information and the computational cost should be considered. After the valid distance is defined based on SVM with an initial sample set, virtual sample candidates are generated from two opposite samples within the valid distance. The first pair of virtual samples is the pair between a sample with the smallest absolute function value and its closest opposite signed sample, since they provide the most accurate information. Next pair is the candidate which has the longest distance from both real and previously selected virtual samples to prevent clustered virtual samples within a small region. Virtual samples are selected and added until new virtual samples are close to previously chosen virtual samples. In Fig. 5, suppose that pair 2 is generated from the smallest absolute value. Then, pair 2 is the first pair of virtual samples. Pair 4 has the longest distance from pair 2 and thus pair 4 is selected as virtual samples next. By applying this method, virtual samples are located uniformly in the design space. Once all virtual samples are selected, the VSVM decision function is constructed by using both existing and virtual samples.

As explained in Section 3.1, true function values at location of "zero point" and virtual samples are not evaluated. Their signs are decided by approximations. Even though signs and locations of virtual samples may not be accurate with initial samples, the accuracy is improved as new samples are inserted sequentially.

3.2 Adaptive strategy for sampling and stopping criteria

3.2.1 Adaptive sequential sampling

The surrogate-based methods construct a model that is accurate over the given domain, and thus samples tend to spread out on the given domain to satisfy the target accuracy. However, since only an accurate decision function is required for the sampling-based reliability analysis and RBDO (Lee et al. 2011), samples near the limit state function are more informative than samples far away from the limit state function. Such efficiency cannot be achieved by using a uniform sampling strategy, and thus a sequential sampling method is proposed for better efficiency and accuracy.

In this paper, new samples are selected such that they are located within the margin ($|s(\mathbf{x})| < 1$), which is narrow since each pair of virtual samples are closely located near the decision function. In addition, new samples should have the maximum distance from the closest existing sample to maximize additional information from new samples. This strategy is similar to the sequential sampling method developed by Basudhar and Missoum (2010), but the computational burden can be reduced by using the within-the-margin (i.e., inequality) constraint ($|s(\mathbf{x})| < 1$) rather than the on-the-decision-function (i.e., equality) constraint ($s(\mathbf{x}) = 0$), which is more difficult to satisfy. On the other hand, this new constraint is effective since the virtual margin is very narrow compared with conventional SVM margins. A less strict constraint can be used with VSVM since new samples do not need to be on the limit state function by introducing virtual samples. In other words, if new samples are located near the limit state function, accurate virtual samples close to the limit state function can be obtained. The optimization problem is defined as

$$\begin{aligned} \max_{\mathbf{x}} \quad & \|\mathbf{x} - \mathbf{x}_{\text{nearest}}\| \\ s.t. \quad & |s(\mathbf{x})| < 1 \end{aligned} \tag{6}$$

where $\mathbf{x}_{\text{nearest}}$ is the existing sample closest to the new sample $\mathbf{x}$. Since $\mathbf{x}_{\text{nearest}}$ changes as the position of new sample candidate $\mathbf{x}$ moves, (6) is a moving target problem. In Fig. 6a, a new sample is positioned by solving (6) and inserted into a region near the limit state function and where there is no existing sample nearby. In Fig. 6b, the VSVM decision function is improved significantly near the newly inserted sample.

Any efficient optimization method can be applied to solve (6). The gradient-based optimization methods such as trust-region-reflective algorithm (Coleman and Li 1996; 1994), active-set algorithm (Powell 1978a, b) or interior-point algorithm (Byrd et al. 2000; Waltz et al. 2006) can be used instead of the PS method since they are faster than the PS method (Lewis and Torczon 1999) without sacrificing much accuracy. In the paper, active-set algorithm

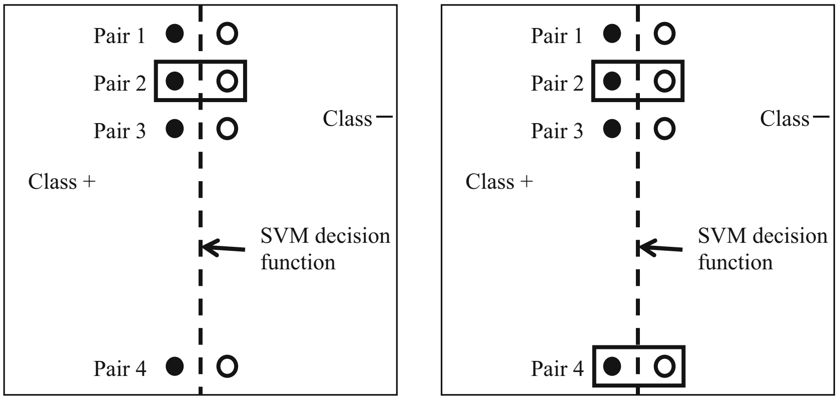
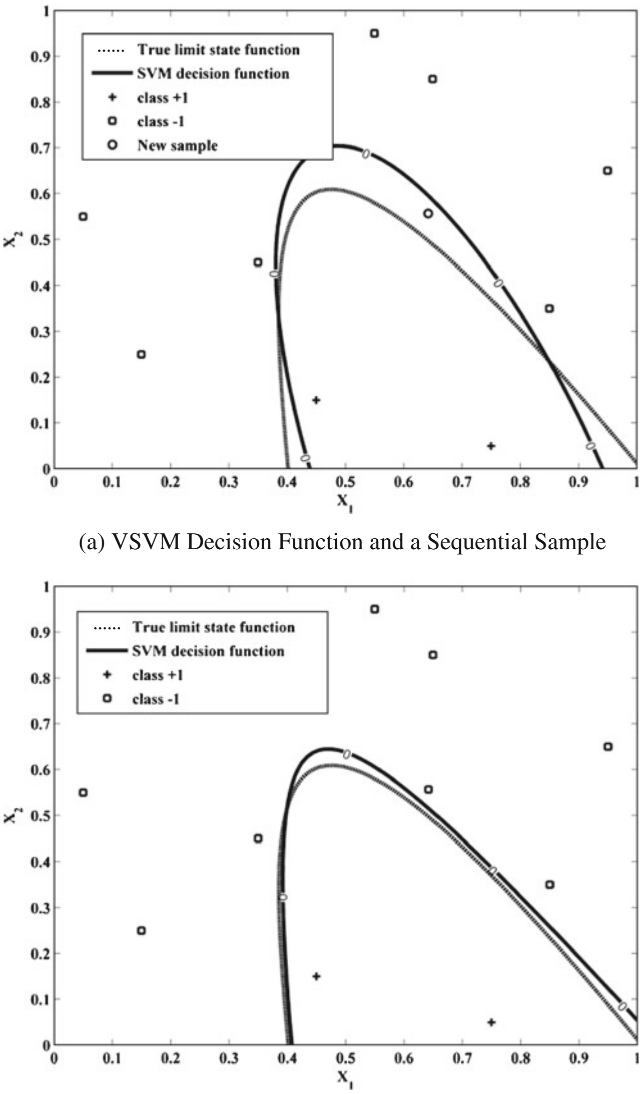


Fig. 5 Selection of virtual samples—pairs within *solid squares* are selected



(a) VSVM Decision Function and a Sequential Sample

(b) VSVM Decision Function with a New Sample

Fig. 6 Changes of the VSVM decision function in normalized design space

is used and initial points are selected among zero points in sparse region.

In the beginning, locations of virtual samples may not be accurate, because their signs are decided based on approximations. However, the accuracy is improved as new samples are inserted sequentially near the decision function.

3.2.2 Stopping criteria

Stopping criteria are required to determine when the decision function is converged. Since the true limit state function is not known, the criterion is based on the variations of the approximated decision function. A set of testing points is generated using input distributions because the MCS points are also generated in the same way for the sampling-based reliability analysis. In this paper, ten thousand testing points for stopping criteria (N_{stop}) were used for all examples. The fraction of testing points that show different signs from the previous surrogate is calculated as (Basudhar and Missoum 2008)

$$\Delta_k = \frac{\sum_{i=1}^{N_{\text{stop}}} I_k(\mathbf{x}_i)}{N_{\text{stop}}} \times 100\,(\%) \tag{7}$$

where k is the current iteration number in the sequential sampling process, and Δ_k is the fraction of testing points for which the sign of the VSVM evaluation changes between k-1th and kth iterations. $I_k(\mathbf{x}_i)$ in (7) is an indicator function defined as

$$I_k(\mathbf{x}_i) \equiv \begin{cases} 0, & sign\,(s_{k-1}(\mathbf{x}_i)) = sign\,(s_k(\mathbf{x}_i)) \\ 1, & \text{otherwise} \end{cases} \tag{8}$$

where $s_{k-1}(\mathbf{x}_i)$ and $s_k(\mathbf{x}_i)$ represent the VSVM value at $\mathbf{x}_i$ at the k-1th and kth iterations, respectively. Changes in the VSVM decision function fluctuate and usually decrease as the number of iterations increases, as is shown in Fig. 7.

In order to implement more stable stopping criteria, the fraction of testing points changing signs between successive iterations is fitted by an exponential curve as Basudhar and Missoum (2008)

$$\widehat{\Delta}_k = Ae^{Bk} \tag{9}$$

where $\widehat{\Delta}_k$ represents the fitted values of Δ_k, and A and B are the parameters of the exponential curve. The value of $\widehat{\Delta}_k$

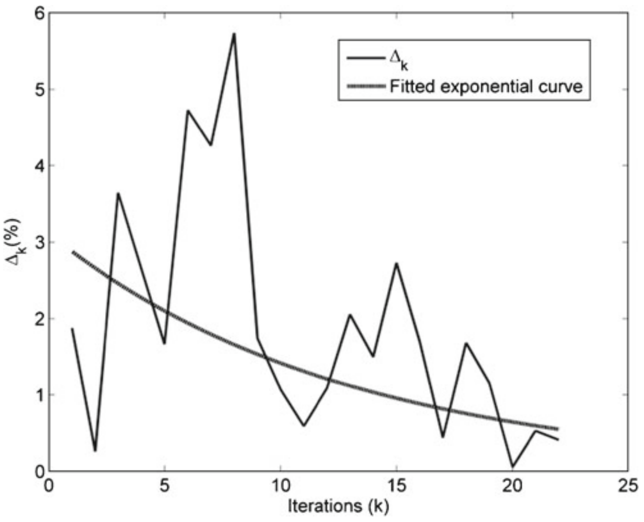


Fig. 7 Changes of Δ_k and fitted exponential curve

and the slope of the curve are calculated when new samples are added. If Δ_k is large while $\widehat{\Delta}_k$ is small, it means that a large change occurred in the model at kth iteration, which $\widehat{\Delta}_k$ did not catch. If Δ_k is small while $\widehat{\Delta}_k$ is large, the situation is that the new sample is inserted into a region where zero-position approximations are already accurate, so there is a small change between recent two models but it may not be converged yet. Therefore, both Δ_k and $\widehat{\Delta}_k$ should be kept small for more robust results. In addition, the slope of the curve needs to be small for stable convergence.

To stop the updating process, the maximum of Δ_k and $\widehat{\Delta}_k$ should be less than a small positive number ε_1. Simultaneously, the absolute value of the slope of the curve at convergence should be lower than ε_2. Thus, the stopping criteria can be defined as

$$\begin{aligned} &\max\left(\Delta_k, \widehat{\Delta}_k\right) < \varepsilon_1 \\ &-\varepsilon_2 < BAe^{Bk} < 0. \end{aligned} \tag{10}$$

where ε_1 and ε_2 are determined so that the target classification error level can be achieved. The target classification error is 2.0 % in this paper. For more accurate limit state function, smaller values can be used. Generally, ε_2 should be smaller than ε_1 for stable convergence.

The overall procedure of the proposed VSVM method with the sequential sampling strategy is shown as Fig. 8.

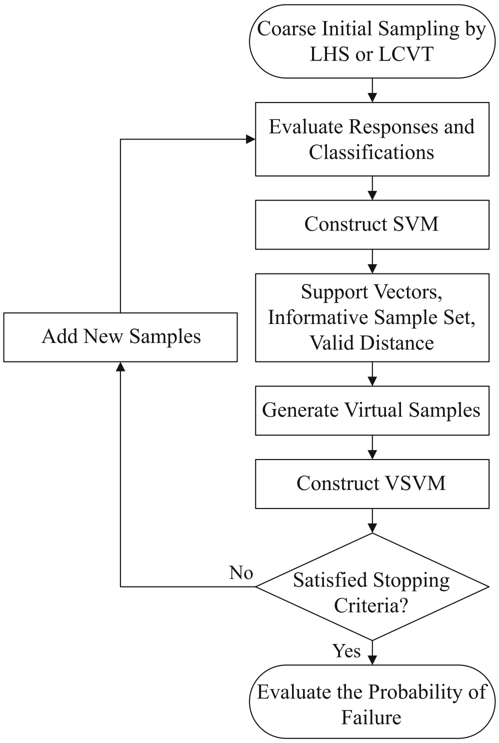

Coarse Initial Sampling by LHS or LCVT
Evaluate Responses and Classifications
Construct SVM
Support Vectors, Informative Sample Set, Valid Distance
Generate Virtual Samples
Construct VSVM
Satisfied Stopping Criteria?
No
Yes
Add New Samples
Evaluate the Probability of Failure

Fig. 8 Flowchart of VSVM with sequential sampling strategy

4 Comparison study between VSVM and other surrogates

4.1 Comparison procedure

Recently developed classification and surrogate modeling methods with sequential sampling schemes are selected for comparison with the proposed VSVM method. The classification method is the explicit design space decomposition (EDSD) method with an improved adaptive sampling scheme that is based on the conventional SVM (Basudhar et al. 2012; Basudhar and Missoum 2008, 2010). The adaptive sampling method of EDSD has two kinds of methods to select new samples. One method is to select new samples which maximize the distance to the closest existing samples while lying on the SVM decision function. The other method is to increase the convergence and selects new samples in a region where data from one class is sparse in the vicinity of the boundary. For a fair comparison for both EDSD and VSVM, the same Gaussian kernel parameter σ in (3) is used. We have exactly implemented EDSD following Refs. Basudhar and Missoum (2008, 2010). Authors of Refs. Basudhar and Missoum (2008, 2010) also applied their EDSD to the same examples in this paper under the same condition and their results are similar to our results.

The surrogate modeling method is the dynamic Kriging (DKG) method with a sequential sampling method (Zhao et al. 2011). Zhao et al. showed that, when the same number of samples is used, DKG performs better compared with the polynomial response surface, radial basis function, support vector regression, and universal Kriging method. Therefore, DKG is chosen to compare the accuracy of VSVM. The basic form of the DKG prediction is expressed as

$$\widehat{y}(\mathbf{x}) = (\mathbf{r}_0 - \mathbf{F}\lambda)^T \mathbf{R}^{-1} y \tag{11}$$

where $\mathbf{R}$ is the symmetric correlation matrix, $\mathbf{r}_0$ is the correlation vector between the prediction location $\mathbf{x}$ and all N samples $\mathbf{x}_i$, $i = 1, \cdots, N$, $\mathbf{y}$ is the response vector, $\mathbf{F}$ is a design matrix of basis functions, and $\boldsymbol{\lambda}$ is a regression coefficient vector. In the DKG method, $\mathbf{F}$ is not fixed, but the best basis function set is chosen by the genetic algorithm (GA) (Zhao et al. 2011). The sequential sampling method chooses new samples where the prediction variance is largest. $\mathbf{R}$ is calculated implicitly and thus DKG is an implicit surrogate method.

Three examples are used to test the performance of the adaptive sampling-based VSVM. One example is a low-dimensional problem, and the other two are high-dimensional problems. All three methods can be applied to both global and local windows. However, a global window usually requires unnecessarily many samples to achieve the target accuracy in RBDO compared with local windows.

Therefore, all three methods are applied to local windows of the original input domain. Since classification methods can be used when the limit state function exists within the local window, original functions are shifted appropriately to include both signed samples. In Sections 4.2, 4.3, and 4.4, the domains of interest are hyper-cube local windows, which are defined by the lower and upper bounds.

For the Gaussian kernel in (3), parameter σ should be provided. Selection of optimum σ is an ongoing research subject. In this paper, a fixed σ value, which is small enough to maintain zero training error, is used. The training error is defined as the classification error with respect to the existing and virtual samples and not testing samples.

Since SVM is a classification method and only takes care of the decision function, the mean squared error (MSE) or R^2, which are widely used for the surrogate-based methods, is not appropriate for comparison. Therefore, the accuracy of the SVM decision function should be judged by its closeness to the true limit state function. In practical applications, the true limit state function is not known, and so is the error measure. However, the error measure can be obtained for academic analytical test problems. One million testing points for error measure (N_{test}) are generated based on input distributions because the MCS samples are also generated in the same way for the sampling-based reliability analysis. These testing points are used to calculate the classification error, which is the fraction of misclassified testing points over total number of testing points. A test point for which the sign of VSVM does not match the sign provided by the true limit state function is considered as misclassification (Basudhar and Missoum 2008). Therefore, the classification error c is

$$c = \frac{\sum_{i=1}^{N_{\text{test}}} I\left(\mathbf{x}_i\right)}{N_{\text{test}}} \times 100\,(\%) \tag{12}$$

where $I(\mathbf{x}_i)$ is the indicator function defined as

$$I\left(\mathbf{x}_i\right) \equiv \begin{cases} 1, & s\left(\mathbf{x}_i\right) \cdot y_i < 0 \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

where y_i represents the corresponding classification value (± 1) at $\mathbf{x}_i$, and $s(\mathbf{x}_i)$ is the VSVM approximation at $\mathbf{x}_i$.

Our purpose is to evaluate the probability of failure accurately and efficiently. The relationship between the probability of failure measurement error and the classification error is approximately proportional. Therefore, accurate probability of failure can be obtained by keeping the classification error small. In addition, the classification error represents the accuracy of the obtained limit state function, so the classification error is used as the error measure for comparison in this paper.

4.2 2-D example

The analytic function is Iowa example function (Tu et al. 1999; Youn et al. 2005), which is expressed as

$$\begin{aligned} f\left(\mathbf{x}\right) = {} & 1 + \left(0.9063 \cdot x_1 + 0.4226 \cdot x_2\right)^2 \\ & + \left(0.9063 \cdot x_1 + 0.4226 \cdot x_2 - 6\right)^3 \\ & - 0.6\left(0.9063 \cdot x_1 + 0.4226 \cdot x_2\right)^4 \\ & - \left(-0.4226 \cdot x_1 + 0.9063 \cdot x_2\right) \\ & 4.5 \le x_1 \le 6.5,\ \ 5.5 \le x_2 \le 7.5 \end{aligned} \tag{14}$$

Since the performance is influenced by sample positions, 20 different test cases are used. The number of initial samples, N_i, is 10, and parameters σ, ε_1, and ε_2 are 3, 0.8, and 0.3, respectively, for both EDSD and VSVM. To compare these methods, VSVM is performed first, and DKG and EDSD are applied later using the same number of additional samples, N_a, as VSVM. Each process is forced to stop when it reaches the same number of final samples. Since each method has its own sequential sampling strategy, sample profiles of the final results are different except the 10 initial samples. According to Table 1, which shows averaged values of 20 test cases, EDSD is the fastest, but the classification result is not accurate. This clearly shows that EDSD converges slowly since it does not use the response function values. VSVM is the most accurate and requires a similar amount of time as DKG (33.1 s vs. 35.3 s) for modeling. However, VSVM is about 30 times faster than DKG (1.1 s vs. 32.5 s) in estimating response values at one million MCS points due to its simpler formulation. Better classification error (2.57 % vs. 0.34 %) is due to different sampling strategies.

4.3 9-D example

The nine-dimensional extended Rosenbrock function (Dixon and Szegö 1978) is used for the test, which is expressed as

$$\begin{aligned} f\left(\mathbf{x}\right) = {} & \sum_{i=1}^{8}\left[\left(1 - x_i\right)^2 + 100\left(x_{i+1} - x_i^2\right)^2\right] - 68000 \\ & -3 \le x_i \le -2,\ i = 1, \ldots, 9. \end{aligned} \tag{15}$$

Table 1 Average results for 2-D example ($N_i = 10$, $N_a = 5.6$, 20 test cases)

		DKG	EDSD	VSVM
Classification error (%)		2.57	15.34	0.34
Elapsed time (sec)	Modeling	35.3	3.2	33.1
	MCS	32.5	0.6	1.1

Table 2 Average results for 9-D example ($N_i = 20$, $N_a = 27$, 20 test cases)

		DKG	EDSD	VSVM
Classification error (%)		2.31	6.79	1.78
Elapsed time (sec)	Modeling	196	60	103
	MCS	49.6	1.8	3.4

Table 4 Average results of EDSD and VSVM for the same stopping criteria ($\varepsilon_1 = 0.25$, $\varepsilon_2 = 0.015$, 20 test cases)

		EDSD	VSVM
No. of additional samples		77.9	33.3
Classification error (%)		6.90	1.67
Elapsed time (sec)	Modeling	149	169
	MCS	1.9	4.6

The initial sample size is 20; and 20 different test cases are used. For both EDSD and VSVM, σ, ε_1, and ε_2 are 5, 0.5, and 0.03, respectively. Twenty-seven additional samples are used for all surrogate methods. In Table 2, EDSD requires less time than other methods, but it is not accurate. The VSVM uses about half amount of time as DKG (103 s vs. 196 s) for modeling and results in better classification error. While VSVM is about twice faster than DKG in estimating response values for MCS at one million sample points, VSVM is about 15 times more efficient than DKG (3.4 s vs. 49.6 s).

4.4 12-D example

For a twelve-dimensional example, the Dixon-Price function (Dixon and Szegö 1978) is used and its mathematical expression is

$$f(\mathbf{x}) = (x_1 - 1)^2 + \sum_{i=2}^{12} i\left(2x_i^2 - x_{i-1}\right)^2 - 36000$$
$$3 \le x_i \le 4,\ i = 1, \ldots, 12. \qquad (16)$$

is used. The initial sample size is 35 for 20 different test cases. Parameters σ, ε_1, and ε_2 are 15, 0.25, and 0.015, respectively. Thirty-three additional samples are used for all three methods. In Table 3, EDSD is very efficient but does not provide accurate results. The VSVM uses less time than DKG for both modeling and estimating response values for MCS and results in a better classification error.

For another way of comparison, EDSD is performed using the same stopping criteria as VSVM so that EDSD can use more samples to construct the decision function. According to Table 4, the average number of additional samples of EDSD is 77.9, which is far more than 33.3 of VSVM. The EDSD also uses a similar amount of time as VSVM, and the classification error is still quite large. Clearly, VSVM is more accurate than EDSD.

Since DKG and VSVM use different stopping criteria, a smaller stopping criterion is used for DKG to achieve a classification error similar to that of VSVM. In Table 5, DKG can achieve a classification error level similar to that of VSVM after it uses about six more samples. Furthermore, the elapsed time of DKG is larger than that of VSVM for both modeling (341 s vs. 169 s) and estimating response values for MCS (65.1 s vs. 4.6 s).

In Fig. 9, even though classification errors with initial samples are not satisfactory, they are decreasing as new samples are inserted. Compared with EDSD, the classification error of VSVM is significantly reduced. Both VSVM and DKG are accurate overall and their convergences are also stable. If new samples are inserted into the region which is already accurate, then new surrogate model will be almost identical to previous surrogate model. This is the reason some flat regions exist for all three methods as shown in Fig. 9.

The VSVM is more efficient than DKG in terms of the elapsed time for both surrogate modeling and estimating response values for MCS, while maintaining a better accuracy level. The EDSD converges very slowly and is inefficient in terms of the number of additional samples. This will be more problematic when the computer simulations at each sample point are expensive for practical application problems.

Table 3 Average results for 12-D example ($N_i = 35$, $N_a = 33.3$, 20 test cases)

		DKG	EDSD	VSVM
Classification error (%)		2.02	8.88	1.67
Elapsed time (sec)	Modeling	289	64	169
	MCS	64.3	1.9	4.6

Table 5 Average results of DKG and VSVM for similar classification errors (20 test cases)

		DKG	VSVM
No. of additional samples		39.4	33.3
Classification error (%)		1.74	1.67
Elapsed time (sec)	Modeling	341	169
	MCS	65.1	4.6

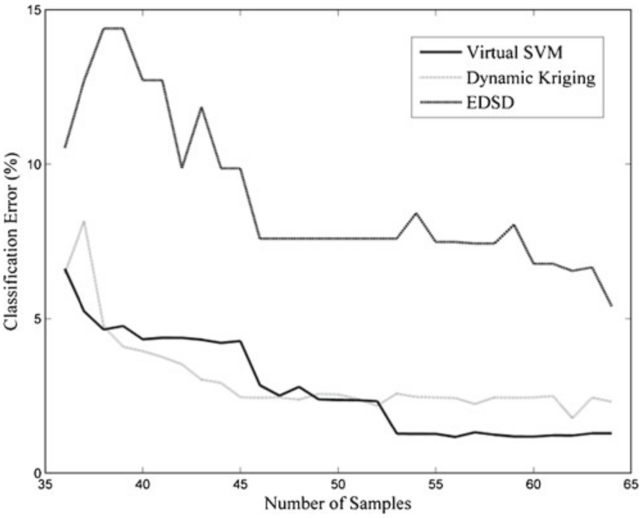


Fig. 9 Classification error changes as VSVM converges

5 Conclusion

A sequential sampling-based virtual support vector machine (VSVM) method is proposed to efficiently construct the accurate decision function for the reliability analysis. Virtual samples are generated from real samples to improve the accuracy of the SVM decision function, and a sequential sampling method is developed to increase the efficiency and accuracy of VSVM by inserting new samples near the true limit state function.

The proposed method is compared with a classification method EDSD and a surrogate modeling method DKG with their own sequential sampling strategies. DKG can construct accurate surrogates with a relatively small number of samples, but it is inefficient for MCS since it has an implicit expression for response evaluations, and the dynamic basis selection process requires significant computational effort. For a low-dimensional problem, both VSVM and DKG are accurate and require similar modeling time. However, VSVM becomes more efficient than DKG or EDSD while achieving excellent accuracy for high-dimensional problems. VSVM is much more efficient than DKG in evaluating response values for MCS, and thus VSVM is preferred for sampling-based RBDO. In this comparison study, better classification error of VSVM compared with DKG is due to the fact that VSVM used the new sequential sampling method near the constraint boundary. Therefore, sampling near the constraint is more efficient than sampling on the whole domain; and it is desirable to implement a constraint boundary sampling method (Bichon et al. 2008, 2011; Ranjan et al. 2008; Lee and Jung 2008; Bect et al. 2012; Viana et al. 2012) for DKG for sampling-based RBDO.

Since the proposed method is developed for contiguous and differentiable problems. For disjoint problems, ordinary classification methods are better suited. Since VSVM requires an approximation method, it is desirable to investigate different approximation methods for generating virtual samples. The choice of the best kernel function and its parameter can be investigated, too. The RBDO problem usually has multiple constraints and the current VSVM method requires building VSVM model for each constraint. It would be more efficient if we can construct one VSVM model for multiple constraints.

The proposed method is developed and applied to sampling-based RBDO using local windows. Therefore, whenever the design is changed, active/violated constraints are identified and DKG and VSVM are applied only for the active/violated constraints (Lee et al. 2011; Youn et al. 2005; Zhao et al. 2011). Thus, both DKG and VSVM use the same local windows and same samples on the local windows. Another fact to point out is that response surface methods such as Kriging have advantage that they can describe not only the limit state but also the overall design space. However, only the limit stat information is required for the sampling-based RBDO so the classification method can be very effectively used.

Acknowledgments The research is jointly supported by the ARO Project W911NF-09–1–0250 and the Automotive Research Center, which is sponsored by the U.S. Army TARDEC. The research is also partially supported by the World Class University Program through the National Research Foundation of Korea (NRF) grant funded by the Ministry of Education, Science and Technology (Grant Number R32–2008–000–10161–0 in 2009). These supports are greatly appreciated.

References

Barton RR (1994) Metamodeling: a state of the art review. In: WSC ’94: Proceedings of the 26th conference on winter simulation, anonymous society for computer simulation international, San Diego, CA, USA, pp 237–244

Basudhar A, Missoum S (2008) Adaptive explicit decision functions for probabilistic design and optimization using support vector machines. Comput Struct 86(19–20):1904–1917

Basudhar A, Missoum S (2010) An improved adaptive sampling scheme for the construction of explicit boundaries. Struct Multidisc Optim 42(4):517–529

Basudhar A, Dribusch C, Lacaze S (2012) Constrained efficient global optimization with support vector machines. Struct Multidisc Optim 46(2):201–221

Bect J, Ginsbourger D, Li L, Picheny V, Vazquez E (2012) Sequential design of computer experiments for the estimation of a probability of failure. Stat Comput 22(3):773–793

Bichon BJ, Eldred MS, Swiler LP, Mahadevan S, McFarland JM (2008) Efficient global reliability analysis for nonlinear implicit performance functions. AIAA J 46(10):2459–2468

Bichon BJ, McFarland JM, Mahadevan S (2011) Efficient surrogate models for reliability analysis of systems with multiple failure modes. Reliab Eng Syst Safety 96(10):1386–1395

Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc 2(2):121–167

Byrd RH, Gilbert JC, Nocedal J (2000) A trust region method based on interior point techniques for nonlinear programming. Math Program 89(1):149–185

Canu S, Grandvalet Y, Guigue V (2005) SVM and Kernel Methods Matlab Toolbox. http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/index.html

Cherkassky V, Mulier F (1998) Learning from data: concepts, theory. Wiley, New York

Ching J (2011) Applications of Monte Carlo method in science and engineering. In Tech, Chap 31

Coleman TF, Li Y (1994) On the convergence of reflective newton methods for large-scale nonlinear minimization subject to bounds. Math Program 67(2):189–224

Coleman TF, Li Y (1996) An interior, trust region approach for nonlinear minimization subject to bounds. SIAM J Optim 6:418–445

Cressie NAC (1991) Statistics for spatial data. Wiley, New York

Dixon LCW, Szegö GP (1978) Towards global optimization 2. North-Holland, Amsterdam

Forrester A, Keane A (2009) Recent advances in surrogate-based optimization. Prog Aerosp Sci 45(1–3):50–79

Forrester A, Sobester A, Keane A (2008) Engineering design via surrogate modeling, a practical guide. Wiley, United Kingdom

Haldar A, Mahadevan S (2000) Probability, reliability and statistical methods in engineering design. Wiley, New York

Hsu CW, Chang CC, Lin CJ (2004) A practical guide to support vector classification. Technical Report. http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

Hurtado JE, Alvarez DA (2003) Classification approach for reliability analysis with stochastic finite-element modeling. J Struct Eng 129(8):1141–1149

Jin R, Chen W, Simpson T (2001) Comparative studies of metamodeling techniques under multiple modeling criteria. Struct Multidisc Optim 23(1):1–13

Kecman V (2001) Learning and soft computing: support vector machines, neural networks, and fuzzy logic models. MIT Press, Cambridge

Lee TH, Jung JJ (2008) A sampling technique enhancing accuracy and efficiency of metamodel-based RBDO: constraint boundary sampling. Comput Struct 86(13–14):1463–1476

Lee I, Choi K, Zhao L (2011) Sampling-based RBDO using the stochastic sensitivity analysis and dynamic Kriging method. Struct Multidisc Optim 44(3):299–317

Lewis RM, Torczon V (1999) Pattern search algorithms for bound constrained minimization. SIAM J Optim 9(4):1082–1099

Martin JD (2009) Computational improvements to estimating Kriging metamodel parameters. J Mech Des 131(8). doi:10.1115/1.3151807

Powell MJD (1978a) A fast algorithm for nonlinearly constrained optimization calculations. In: Watson GA (ed) Numerical analysis, Lecture notes in mathematics, Springer Verlag, p 630

Powell MJD (1978b) The convergence of variable metric methods for nonlinearly constrained optimization calculations. In: Mangasarian OL, Meyer RR, Robinson SM (eds) Nonlinear programming 3, Academic Press

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

Rubinstein RY (1981) Simulation and the Monte Carlo method. Wiley, New York

Saka Y, Gunzburger M, Burkardt J (2007) Latinized, improved LHS, and CVT point sets in hypercubes. Int J Numer Anal Model 4(3–4):729–743

Schölkopf B (1999) Advances in Kernel methods support vector learning. MIT Press, Cambridge

Schölkopf B, Smola AJ (2002) Learning with Kernels: support vector machines, regularization, optimization, and beyond. MIT Press, Cambridge

Simpson T, Poplinski J, Koch P (2001) Metamodels for computer-based engineering design: survey and recommendations. Eng Comput 17(2):129–150

Tu J, Choi KK, Park YH (1999) A new study on reliability-based design optimization. J Mech Des 121(4):557–564

Vapnik VN (1998) Statistical learning theory. Wiley, New York

Vapnik VN (2000) The nature of statistical learning theory. Springer, New York

Viana FAC, Haftka R, Watson L (2012) Sequential sampling for contour estimation with concurrent function evaluations. Struct Multidisc Optim 45(4):615–618

Viana FAC (2010) SURROGATES Toolbox User's Guide. http://sites.google.com/site/fchegury/surrogatestoolbox

Waltz RA, Morales JL, Nocedal J, Orban D (2006) An interior algorithm for nonlinear optimization that combines line search and trust region steps. Math Program 107(3):391–408

Wang GG, Shan S (2007) Review of metamodeling techniques in support of engineering design optimization. J Mech Des 129(4):11

Youn BD, Choi KK, Du L (2005) Enriched performance measure approach for reliability-based design optimization. AIAA J 43(4):874–884

Zhao L, Choi KK, Lee I (2011) Metamodeling method using dynamic Kriging for design optimization. AIAA J 49(9):2034–2046