

Classification-based Optimization and Probabilistic Analysis Using LS-OPT[®]

Anirban Basudhar¹, Katharina Witowski², Imtiaz Gandikota¹

¹Livermore Software Technology Corporation, Livermore CA, USA

²DYNAmore GmbH, Germany

Abstract

LS-OPT is a standalone Design Optimization and Probabilistic Analysis package with an interface to LS-DYNA[®] that is capable of solving a variety of reliability assessment and single or multi-disciplinary design problems. It consists of a flexible framework with various methods that are suited to specific types of problems. The solution methods in LS-OPT are broadly categorized as direct or metamodel-based. Metamodel-based methods build approximations of the system responses using only a few samples, before applying core optimization or probabilistic analysis methods on these inexpensive surrogate models. They are particularly attractive due to higher efficiency compared to direct methods, which on the contrary can be prohibitively expensive in some scenarios. However, metamodel-based approximation is hindered in certain situations, such as due to discontinuous or binary responses. Instances of such responses include buckling during impact, aerodynamic flutter, fluid leakage simulation, nonlinear energy sinks etc. To overcome such issues, a classification-based method has been developed, which only uses binary safe/failed (or feasible/infeasible) information to approximate the decision boundary in the design space.

The version 6.0 of LS-OPT will introduce a new "Classifier" entity to provide greater flexibility in handling such problems. Instead of predicting the response value at an unknown design sample, a classifier predicts whether the design will be feasible. The decision boundary separating the feasible and infeasible samples is the zero level isocontour of an inexpensively computable mathematical expression. Clearly, the method is well suited for binary problems with only pass/fail information (e.g. whether a simulation succeeded or whether there was any blood leakage for an implanted stent for aneurysm etc.). For discontinuous responses also, the problem is greatly simplified as the method depends only on the category and not on the value of response at a sample, thus eliminating the need for fitting a discontinuity. The classification-based method is also very useful for system reliability assessment in the presence of multiple failure modes, as it can represent all the modes with a single classifier. Additional applications of classifiers have also been proposed in the field of multi-objective optimization. A statistical learning technique known as Support Vector Machine (SVM) is used to construct the decision boundaries in LS-OPT. This paper gives an overview of the classification capability of LS-OPT along with a few examples where classifiers can be useful, including examples with response discontinuities.

Introduction

Decision-making is an integral part of any design optimization or design reliability assessment process. One can explore several ways to make a decision, e.g. based on experience, physical tests, simulation results etc. or a combination of these results [1]. Physical tests and simulations provide an insight into the performance metrics of the designs under consideration. Different types of decisions are made based on the results, such as whether one design is better than another is, whether a particular design is feasible or not etc. However, obtaining the best design or calculating the failure probability of a design involves numerous calls to potentially expensive performance evaluation functions. Therefore, an important requirement is to improve the efficiency of the decision making process, i.e. to reduce the number of expensive evaluations needed to get the final solution.

Machine learning techniques [2, 3] are often used to aid in the decision-making process in order to avoid the high cost (or computational time). They are able to provide an insight into the system behavior at an unknown design configuration by learning from a relatively small number of expensive evaluations. A widely used approach is to replace the actual expensive function evaluations with evaluations based on metamodels that act

as surrogate models [4]. Various types of metamodels can be trained based on different basis functions and different criteria for the "best" fit, several of which are also available in LS-OPT [5]. Several sampling strategies have also been developed to globally or locally enhance metamodel accuracy, based on specific applications such as deterministic single-objective design optimization, multi-objective optimization, reliability analysis or reliability-based design optimization [6-9].

For most types of design problems, metamodel-based methods have been shown to perform efficiently and accurately. However, the use of metamodels is hampered for certain systems that have discontinuous or binary responses, e.g. buckling, on-off contact, hidden constraints etc. [10, 11]. The approximation of response values is difficult for such systems. However, these problems are enormously simplified upon the realization that many of the decisions during optimization or reliability assessment, e.g. constraint feasibility check, do not actually need prediction of the response values for the different design configurations; instead, only a binary decision based on classification of the designs as feasible or infeasible is required in such instances. It is possible to obtain such a decision boundary using only the classification information of the training samples [10-14]. The key difference is that instead of substituting the response value evaluation, a classifier simply provides a pass/fail decision [15-19]. In order to have the ability to predict such binary decisions, the classifier only needs to learn from the pass/fail information at a few training samples (and not from the response values with potential discontinuities that can throw off the training process based on continuous basis functions). As a result, the classification-based constraint handling approach automatically mitigates issues such as not having a quantitative response value (binary data) [11, 20] or having a discontinuous response [10, 12].

This paper presents the LS-OPT implementation of classifiers and demonstrates their application to define the constraint boundaries for design optimization [14] or reliability assessment [10, 13]. The principal idea is to define a threshold or clustering-based [21] pass/fail criterion that classifies the training samples into two classes, and then to construct an "optimal" decision boundary in the design space that separates the samples belonging to the different classes. Once the decision boundary has been trained, it predicts the feasibility of any arbitrary design configuration without costly evaluations to enable efficient optimization or reliability assessment.

The aim of this paper is to give an overview of a few engineering design applications where the classification-based method can be advantageous, as well as to provide a general idea of how to use the classification-based design feature of LS-OPT. The paper presents several examples consisting of discontinuous and binary responses, multiple failure modes and adaptive sampling to provide some idea about the scope of applying the new classification-based design approach.

Statistical Classification Background

Statistical classification [2, 3, 16, 22] is a subset of machine learning [2, 3] in which the goal is to identify the category to which a new observation belongs, based on prior training data. The data consists of a sample set \mathbf{X} and their labels \mathbf{Y} . The possible labels in the vector \mathbf{Y} are defined beforehand – e.g. three categories in the data implies three types of labels. The statistical classification problem is that of identifying the best pattern in the variable space that defines the boundary between distinct categories or classes. Figure 1 shows an example of three-class classification.

The use of classification algorithms is prevalent in several disciplines, such as image recognition [23], financial decision making [24], spam filtering [25], gene analysis [26] etc. The same concept can also be used in design optimization [14] and reliability assessment [10, 13], for which it is typical to have two classes of data (feasible and infeasible).

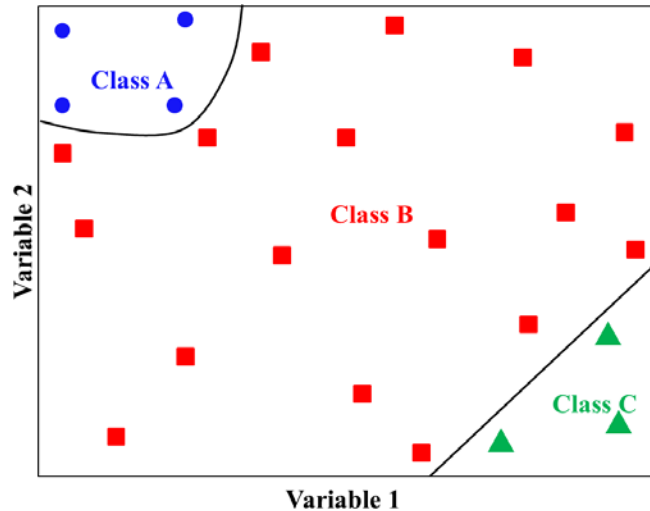


Figure 1. Statistical classification of three distinct category samples (circle, square and triangle).

Some of the common statistical classification algorithms are k nearest neighbors (kNN) [27], support vector machines (SVMs) [18], random forest (RF) [28] and neural networks (NN) [29, 30]. Particular emphasis is given to SVM [18] in this paper, as this algorithm has been implemented in LS-OPT. The basic idea in SVM classification is to construct a linear boundary in a feature space separating two classes of data. Even though the data is not necessarily linearly separable in the input variable space, they can be separated by a hyperplane in this feature space. Additionally, only the inner product of the feature space position vectors is needed, which can be represented by a kernel function in the input space (with some restrictions on the type of kernel function). As a result of this mapping, one can work with the input space variables to construct an optimal boundary in the feature space. The equation of the SVM decision boundary for two-class classification is obtained as $s(\mathbf{x}) = 0$, where $s(\mathbf{x})$ is given in Eq. (1).

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

Here, $y_i = \pm 1$ is the class label, α_i is the Lagrange multiplier for i^{th} sample and b is the bias. K is the kernel function that maps the design space and the feature space (a high-dimensional space consisting of basis functions as the variables) [18]. An SVM constructs the optimal boundary that maximizes the margin between two sample classes (± 1) in a feature space. The boundary is linear in the feature space, but can be highly non-linear in the design space. The margin is defined as the perpendicular distance between $s(\mathbf{x}) = -1$ and $s(\mathbf{x}) = 1$ in the feature space. The SVM is constructed such that there is no training sample within the margin.

Support Vector Machine Classification-based Design Using LS-OPT

LS-OPT 6.0 will introduce a new design framework in which the constraints are defined as SVM classifiers. In both design optimization and reliability assessment one of the main tasks is to differentiate between acceptable (feasible/safe) and unacceptable (infeasible/failed) designs. In optimization, the optimum design is located in the feasible space. Similarly, in reliability assessment, the failed samples contribute to the failure probability. Statistical classifiers are perfectly suited to construct a decision boundary that separates the two classes of data. Once trained, a classifier can predict the class (feasible/infeasible) of a new design configuration almost instantaneously, and is therefore well suited for use within an optimization or reliability assessment framework.

A statistical classifier performs equally well for continuous functions and for discontinuous or binary responses, as it only uses pass/fail information at a few specified samples for training. This information is readily available even for problems with binary or discontinuous responses [12, 14, 20]. The classifier training or learning

process involves finding patterns in the design space that are capable of separating the samples belonging to the feasible and infeasible categories. This is in contrast to metamodeling, where the training involves a functional based on the response values. The difference between metamodel-based and classification-based methods to determine acceptability of any general design alternative is shown in Figure 2. The classification-based method takes a decision directly based on the position of the new sample in the design space, as the classifier is trained to find patterns in the design space, whereas in the metamodel-based method, the decision is taken based on the corresponding predicted response value and threshold.

In general there are infinite patterns that can separate the feasible and infeasible category samples. However, the decision boundary is constructed such that it optimally separates the acceptable and unacceptable training samples using SVM.

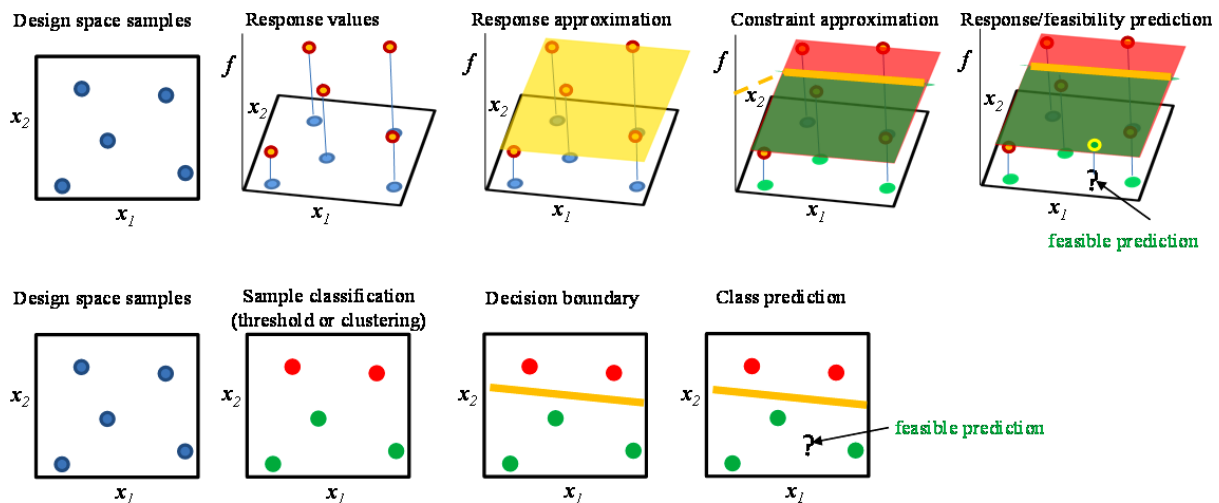


Figure 2 Summary of basic classification method (bottom) and comparison to metamodeling (top).

In LS-OPT one can specify the use of classifiers in a problem in the add menu, which adds a “Classifier” box to the GUI flowchart. The definition of the classifier(s) is then specified in the classifier dialog (Figure 3). Three items are needed to complete the classifier definition:

1. The component(s) of the classifier along with the corresponding classification criterion, e.g. threshold or clustering
2. The type of system (series/parallel) in the case of multiple classifiers
3. The classification algorithm (SVM) and its parameter selection preferences

Once a classifier is defined, it can be used as a constraint in optimization or in Monte Carlo analysis. It is also possible to have a nested classifier definition to get a mixed series-parallel system. For example, one can define classifier 1 and classifier 2 as series and then define a third classifier that has classifier 1 and 2 as parallel components.

The classification criterion based on a pre-defined threshold is quite straightforward, as the class of any design can be determined by comparing its response value to the threshold. However, the threshold is not known a priori always. In such cases, clustering is performed, which falls under the category of unsupervised machine learning [16]. The class of a sample is then determined not just by its response, but by the response of all the samples together. The k-means clustering algorithm is currently available in LS-OPT for this purpose.

An SVM classifier can be trained with several types of kernels. LS-OPT currently allows Gaussian, polynomial and exponential kernels. Additionally, it gives one the option to use fixed kernel parameters or to let LS-OPT select the parameters based on an error rate.

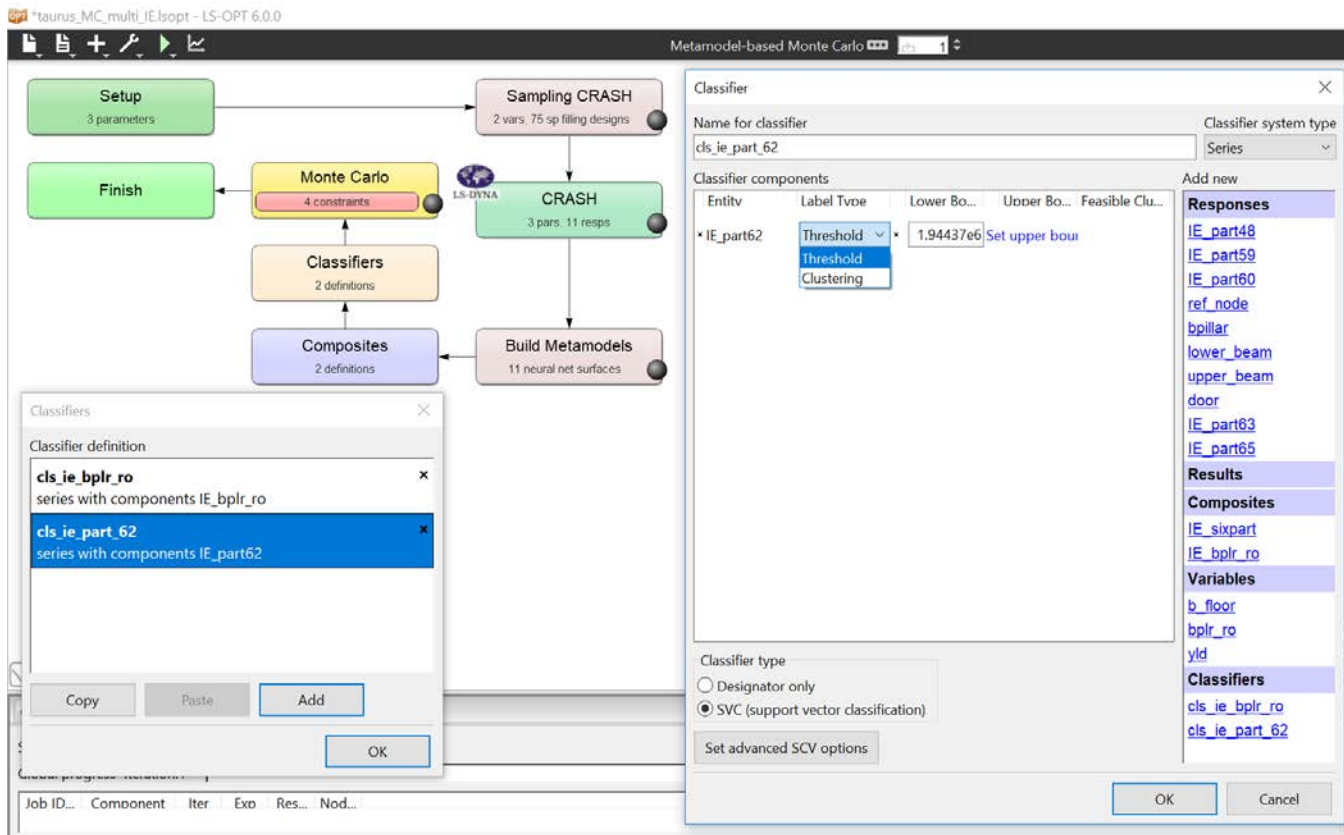


Figure 3. Classifier definition dialog in LS-OPT.

The use of classifiers to define constraint boundaries has several advantages that are presented through examples in the next section. The main advantages of using the classification-based method are also listed below.

1. As evident in Figure 2, the classification-based method does not require the actual response values at the training samples; instead it only requires the class ± 1 of the samples. As a result, it can be applied to problems with only binary pass/fail information available (Figure 4Figure). For example, response evaluation may not be possible for a certain category of infeasible design configurations. In other instances, it is possible to evaluate the responses for all samples, but only as a qualitative indicator of a binary pass/fail status [10, 14]. In non-binary problems also, the presence of response discontinuities and noise hampers the accuracy of the metamodel-based approach [10]. Thus, it also automatically handles failed simulations that do not provide a response value.
2. As the response values are not approximated and are instead classified, the method is unaffected by the presence of discontinuities [10, 31].
3. Different failure modes can be represented by a single classifier. This feature can be used to devise a mechanism in which only a few failure modes need to be evaluated at a sample to obtain the class label, thus increasing the efficiency [14].
4. Because only the classification information is required, it is possible to terminate a simulation before the end time in order to save time.

Examples: Statistical Classification in Design

As mentioned in the previous section, statistical classification has some obvious advantages compared to metamodeling when defining design constraint boundaries, especially for non-smooth responses. This section presents a few examples where the use of classifiers can be useful or even necessary in some instances.

Example 1: Failure Boundary Definition for Binary Response – Ball Impacting Plate

This example consists of a ball impacting a plate modelled with solid elements and the MAT24 material card in LS-DYNA (Figure 4). The plate elements erode upon reaching a specified limiting plastic strain of 0.5. Upon reaching the limit state, an element fails and undergoes an instantaneous drop in strain to zero value. The exact limit state may or may not be captured for a particular simulation in the LS-DYNA output databases, depending on the output intervals. As a result, only pass/fail information is available in the message file. This example is studied with only two variables, plate thickness and the yield stress, for simplification of the visual interpretation. An experimental design consisting of 50 space filling samples is generated. The corresponding binary states represented by +1 (no erosion) and -1 (element erosion) are shown in Figure 4 right. In Figure 5, the metamodel-based approach is compared to the classification-based approach. Metamodels have problems while approximating such binary or discontinuous responses and are sensitive to the selection of arbitrary numeric labels. However, this is a very simple problem from a classification perspective. The SVM boundary can be used for optimization or for reliability assessment.

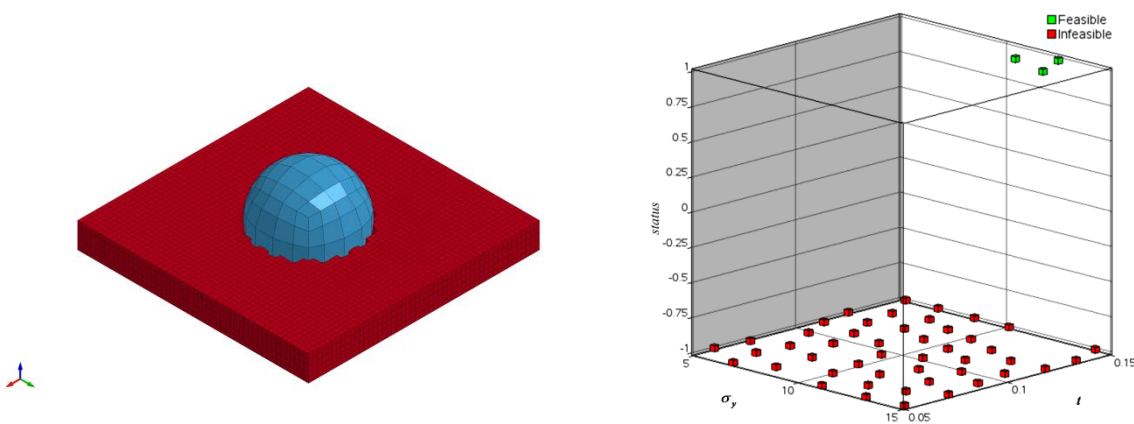


Figure 4: Element erosion upon impact on plate (left). Binary response (right). -1 shows erosion.

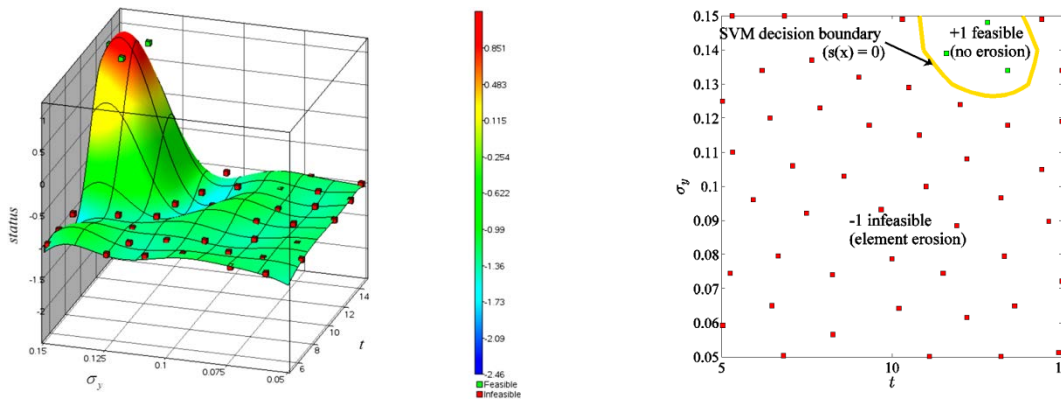


Figure 5: Fitting metamodel to binary response (left). SVM classification of the designs (right)

Example 2: Design Optimization with Mode Tracking (Discontinuous Frequency Constraint)

This example consists of an LS-DYNA simplified car model (Figure 6) for which the mass is minimized while constraining the first torsional mode to be greater than a certain limit. The thickness values of the bumper (tbumper) and the rail (trailb) are the optimization variables. Thus the optimization problem is:

$$\begin{aligned} & \min_{x_1, x_2} \text{Mass} \\ & \text{s.t. } v_{torsion} > 220 \end{aligned} \tag{2}$$

where $v_{torsion}$ is the first torsional mode frequency, x_1 and x_2 are the thicknesses of bumper and rail, respectively. The first torsional mode for the baseline design was identified manually before automatically detecting the closest mode shape for the other samples, based on the modal assurance criterion (MAC). As the torsional mode is tracked, the mode number can switch from one design to another. This is also typically accompanied by a sudden jump in the frequency, leading to a response discontinuity (Figure 7). The problem is solved using a feedforward neural net (FNN) approximation of the frequency as well as with SVM classification and the results are compared. Figure 8 provides the comparison between the metamodel and classification-based constraint boundary approximation. The FNN-based approximation clearly misclassifies some of the feasible training samples and predicts them as infeasible. This leads to a conservative design with $t_{bumper} = 4.71496$ and $trailb = 5.20927$. The corresponding scaled mass objective is 1.78755. Using an SVM-based constraint, the optimal design is located at ($t_{bumper} = 4.42104$, $trailb = 5.06372$) with a lower mass value of 1.74285.

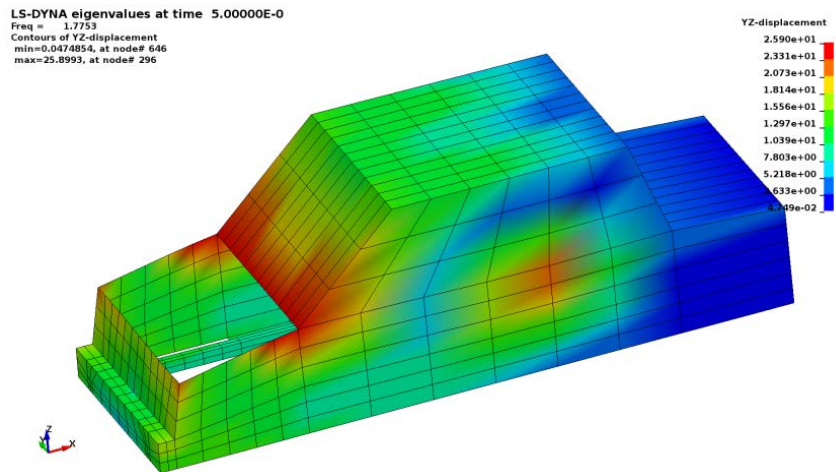


Figure 6 Simplified car LS-DYNA model for modal analysis

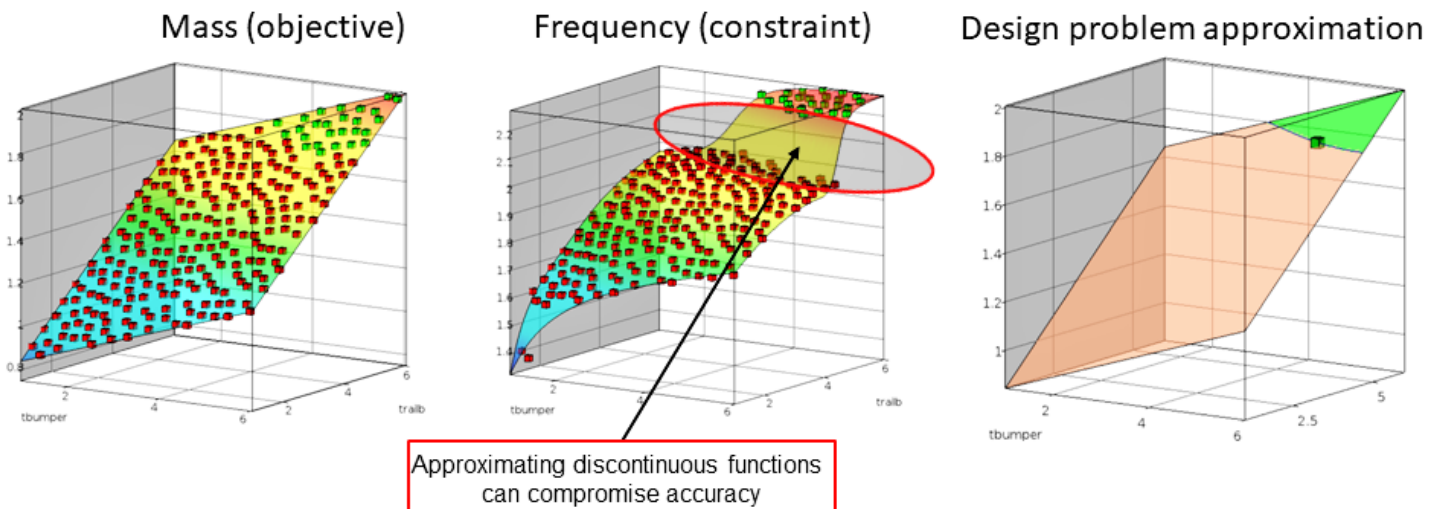


Figure 7. Modal Analysis Example. FNN approximation built for mass minimization with torsional frequency constraint. The frequency response is discontinuous due to mode switching.

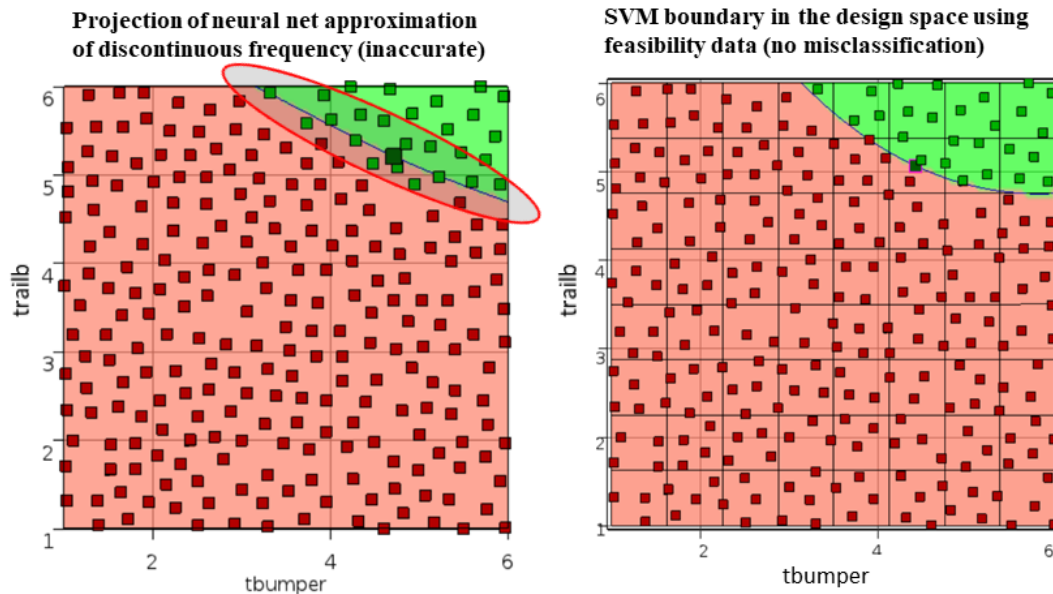


Figure 8. Modal analysis example with discontinuous frequency constraint. Comparison of FNN and SVM constraint boundaries.

Example 3: System Reliability Assessment for Side Impact Noisy Intrusion Constraint

The following example consists of a Monte Carlo analysis of a vehicle impacting a pole. The LS-DYNA finite element model of the vehicle used in the study is shown in Figure 9. The model [32] consists of approximately 30,000 elements and therefore, it is a generalized model intended only for demonstration of application of optimization concepts discussed in this study. The majority of the elements in the model were defined using shell section with ELFORM 2 formulation of LS-DYNA. The finite element model also consisted of a rigid cylindrical wall with a radius of 150mm. The vehicle model was subjected to a side impact against the rigid cylindrical wall at a velocity of 29 mph.

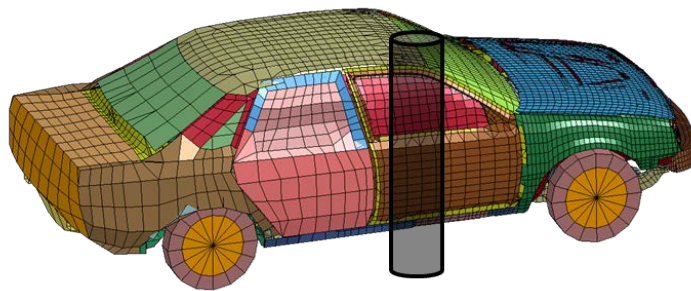


Figure 9: Finite element model of the vehicle-pole side impact analysis

The shell thicknesses of a few design parts, selected based on their contribution to overall internal energy absorption, were selected as the design variables for the Monte Carlo analysis. To simplify the setup into a two variable example to enable visualization, the thickness of the b-pillar and the lower beams was defined as variable 1 and the thickness of the two floor parts was defined as variable 2. The intrusion distance of b-pillar, lower beam, and the door were defined as the design constraints for the Monte Carlo analysis. The intrusion distance was calculated as the difference of displacements of the nodes in the intrusion location and a reference node. The design parts and the location of intrusions are shown in Figure 10.

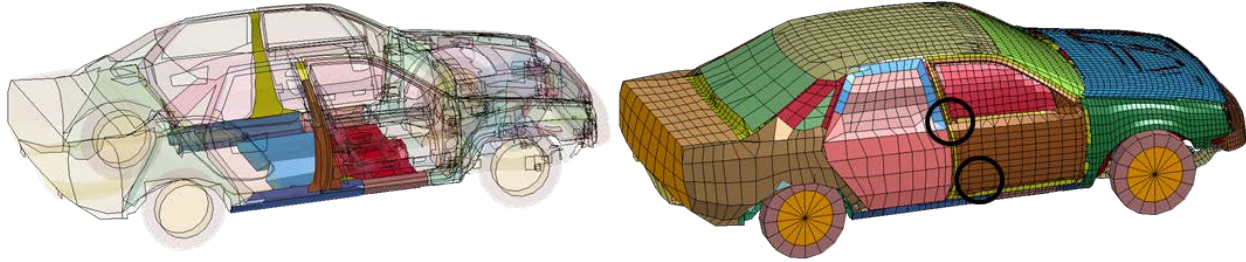


Figure 10: Design parts, shown in darker shade (left), location of b-pillar and door intrusion (right).

In the Monte Carlo analysis, the two noise variables, beam and b_floor, are normally distributed with a mean and standard deviation of 4, 0.4 mm and 2.5, 0.25 mm, respectively. The intrusion distances were defined as the design constraints to determine the probability of exceeding the constraint bounds using Monte Carlo analysis. The probability of failure for the system is defined as the probability of exceeding the constraint bound of any of the intrusion constraints, given as:

$$\text{Failure Probability } P_f = P [(g_1 > 585\text{mm}) \cup (g_2 > 710\text{mm}) \cup (g_3 > 638.23\text{mm})] \quad (3)$$

where g_1 , g_2 , g_3 are the intrusion distances of the b-pillar, lower beam, and the door.

In order to get an insight into the nature of the intrusion responses, FNN metamodels were constructed and visualized. The response approximations and simulation points are plotted in Figure 11 top, which shows the presence of noise in the constraint functions.

The metamodel-based Monte Carlo framework of LS-OPT was used to calculate the failure probability using the FNN metamodels. The corresponding predicted failure region is shown in Figure 11 bottom left. It is quite clear that the quality of the approximation is lacking in the vicinity of the failure boundary, as a result of which 5 of the training samples are misclassified among 75 (6.67 % training misclassification).

In order to compare the metamodel-based and classification-based approaches, a single SVM decision boundary was constructed using the same 75 samples to represent all three constraints. The SVM kernel parameters were selected automatically by LS-OPT, using five-fold cross-validation to minimize the error rate. The SVM decision boundary (Figure 11 bottom right) correctly classifies most of the samples, except for one. The system failure probabilities using the FNN metamodel and the SVM classifier are compared in Table 1. The failure probability obtained using 3000 Latin Hypercube finite element runs (direct MC) is also listed. The 95% confidence intervals of the failure probabilities are shown. Both the SVM and FNN based probabilities fall within the 95% confidence interval of the direct MC failure probability value, but the mean values of the direct MC and the SVM estimate coincide almost exactly. Therefore, although it is difficult to provide a conclusive decision due to the relatively large confidence interval of the direct MC, based on the mean failure probability the SVM-based method seems to outperform the FNN metamodel. Visually also, as well as based on the training misclassification, the SVM failure boundary seems to be much more intuitive and accurate. In the future we will need to conduct a direct MC analysis with much higher number of samples to provide a more precise comparison.

Table 1: Training error and 95% failure probability intervals of Ford Taurus side impact intrusion

Method	FNN	SVM	Direct MC
Training error	6.67%	1.33%	-
Failure Probability 95% CI	0.0221 _± 0.0029	0.0237 _± 0.0030	0.0237 _± 0.0094

Another point to note is that failure probability calculation for parallel or mixed systems is not supported using metamodels in LS-OPT. Using a classifier-based constraint one can now define any type of system failure criterion (series/parallel/mixed). This is useful when there is redundancy in the system and as long as one of the redundant units is functional, the system will remain functional.

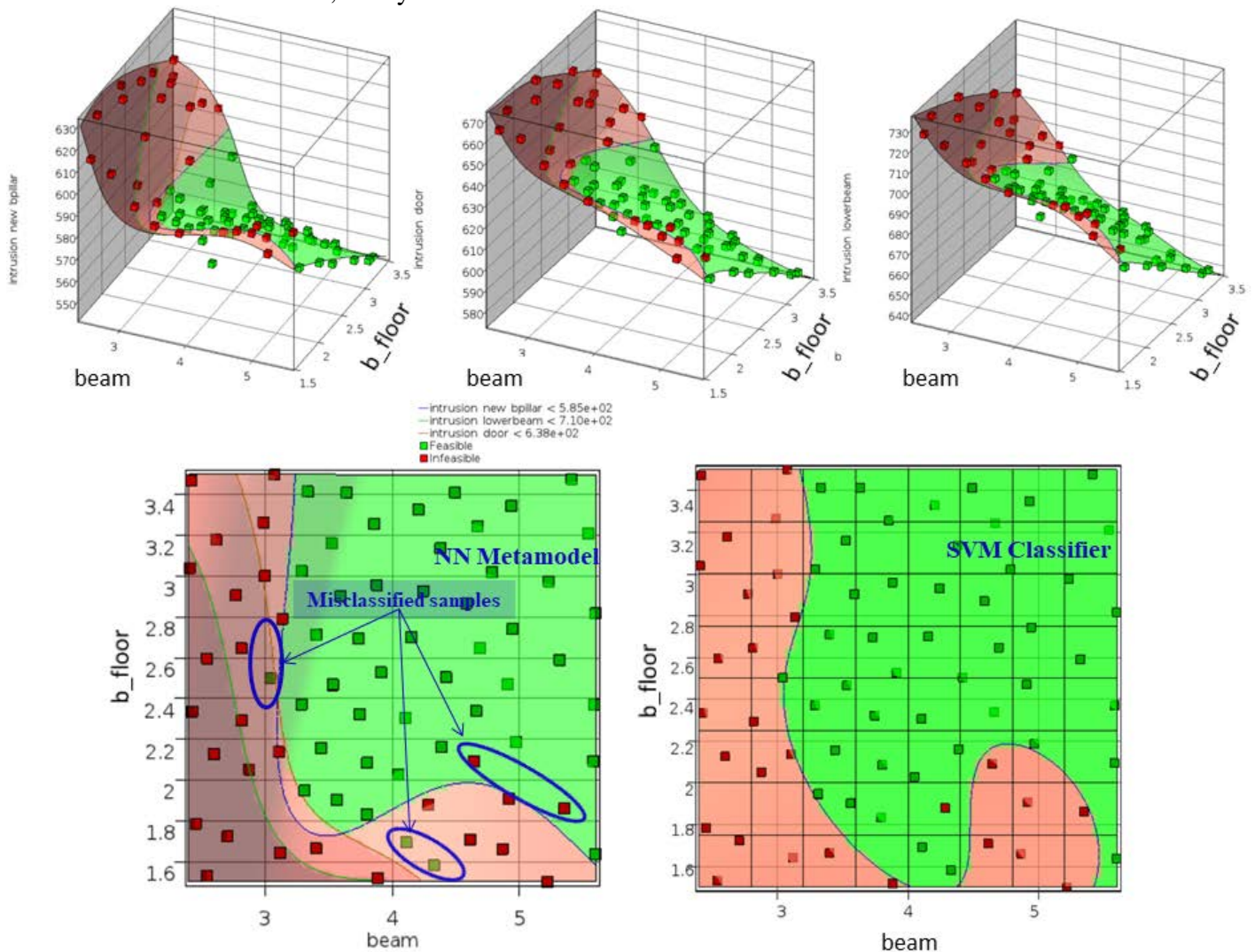


Figure 11. Ford Taurus example with noisy response. FNN response approximations are shown on the top for the three intrusion constraints. The bottom figures depict the failure boundary approximation in the design space using FNN (left) and SVM (right).

Example 4: Adaptive Sampling with SVM-based sampling constraint

One can also use classifiers to define the regions of the design space that need to be sampled further. In many scenarios, sampling large portions of the design space does not bring any useful information [31-33]. For instance, the designs corresponding to those regions may be non-physical or the simulation may fail for those parameter combinations. One may also want to restrict the sampling space simply because the optimum solution is likely to be in a specific part of the design space. Such restrictions can be enforced using a classifier-based sampling constraint. Figure 12 shows one such instance of adaptive sampling with SVM-based sampling constraint, where the goal was to sample in the vicinity of the failure boundary to accurately estimate it. The sample set in the figure is the result of eight iterations with ten samples each. The samples are constrained to lie within the margin of the SVM by defining a sampling constraint in LS-OPT. The SVM margin adapts itself as more samples are added.

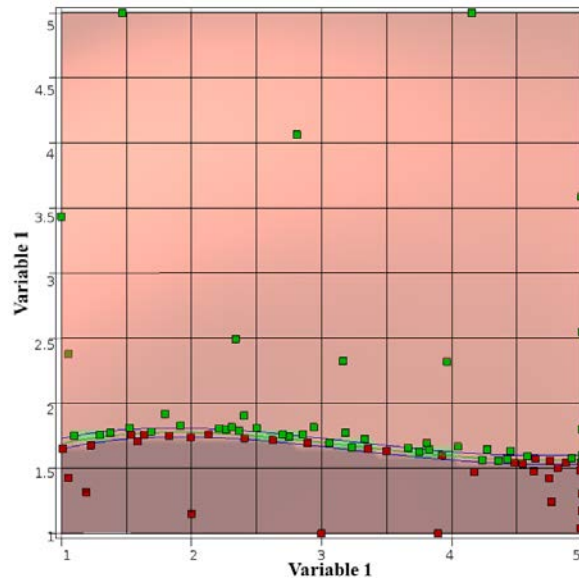


Figure 12. Adaptive sampling using SVM sampling constraint in the vicinity of the decision boundary.

Another special case of SVM-based adaptive sampling is Adaptive Explicit Multi-Objective Optimization (AEMOO) [34]. However this algorithm is not implemented yet in LS-OPT. Nevertheless, the basic concept of AEMOO is presented here with an example. It is indeed possible to set up the AEMOO algorithm in LS-OPT using a multilevel framework, but that would require significant expertise on part of the user.

In general, MOO problems consist of two or more objectives that are maximized or minimized simultaneously and are conflicting in nature. As a result, the solution of a MOO problem is typically a set of Pareto-optimal points instead of a single solution. A part of the design space is Pareto-optimal whereas the other regions are dominated. Thus, MOO can be seen as a binary classification problem in which the two classes are dominated (-1 class) and non-dominated or ND (+1 class) (Figure 13). Once a classifier (e.g. SVM) separating the dominated and non-dominated samples are trained, Pareto-optimality of any design is determined using a single classifier evaluation, in contrast with all existing methods.

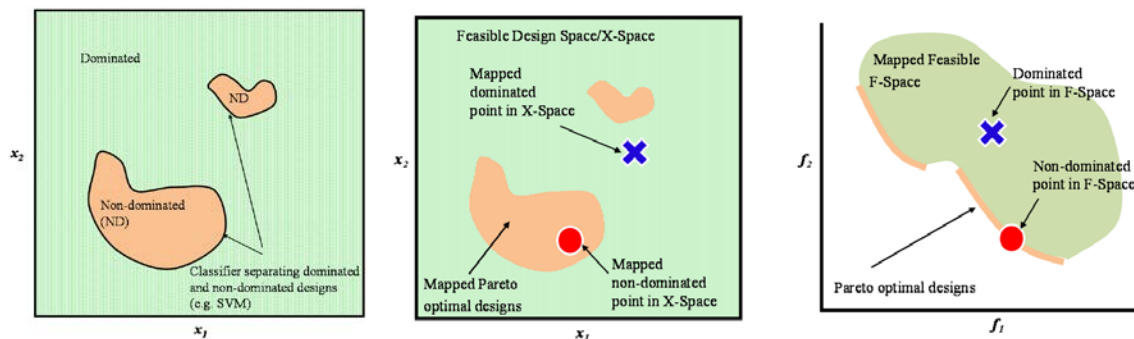


Figure 13: Boundary defining non-dominated regions in design space (left). Design-Objective space Mapping (right).

In order to train an SVM, each training sample's class needs to be assigned prior to the construction of the decision boundary. An issue in using classification for MOO is that although it is known that dominated samples (class -1) cannot be Pareto-optimal, the opposite is not true; being non-dominated among current samples isn't sufficient to be Pareto-optimal. However, the +1 samples represent Pareto-optimal designs if the data is sufficient. A decision boundary obtained by assigning +1 class to the current non-dominated samples represents an estimate of the Pareto-optimal front, and is refined adaptively. As points are added, samples may switch from +1 to -1 as non-dominated samples may be dominated by newer samples until convergence. As class definition of existing samples can change during the course of AEMOO, the classifier is referred to as *dynamic*.

An example of two objective optimization with 30 variables is presented here. The objective functions f_1 and f_2 are:

$$f_1 = x_1 \quad (4)$$

$$f_2 = r(x)h(f_1(x)r(x)), \text{ where } r(x) = 1 + \frac{9}{m-1} \sum_{i=2}^m x_i, h(x) = 1 - \sqrt{\frac{f_1(x)}{r(x)}}$$

where x_1 and x_2 are the two variables. The Pareto front is constructed using three methods – NSGAI [35], Pareto Domain Reduction (PDR) [36] and the adaptive SVM-based AEMOO [34]. The fronts using the three methods are shown in Figure 14, along with the actual one. AEMOO located all five disjoint regions within first 10 iterations, following which it enhanced the diversity. Both NSGAI and PDR were significantly slower and lacked accuracy. Using population size of 48, direct NSGAI completely missed 2 out of 5 regions. PDR was able to sample 4 of the regions satisfactorily at 40-50 iterations (1880-2350 samples). It found one non-dominated sample close to the fifth region, but not on it. Overall, the result demonstrates the superiority of the SVM classification-based approach.

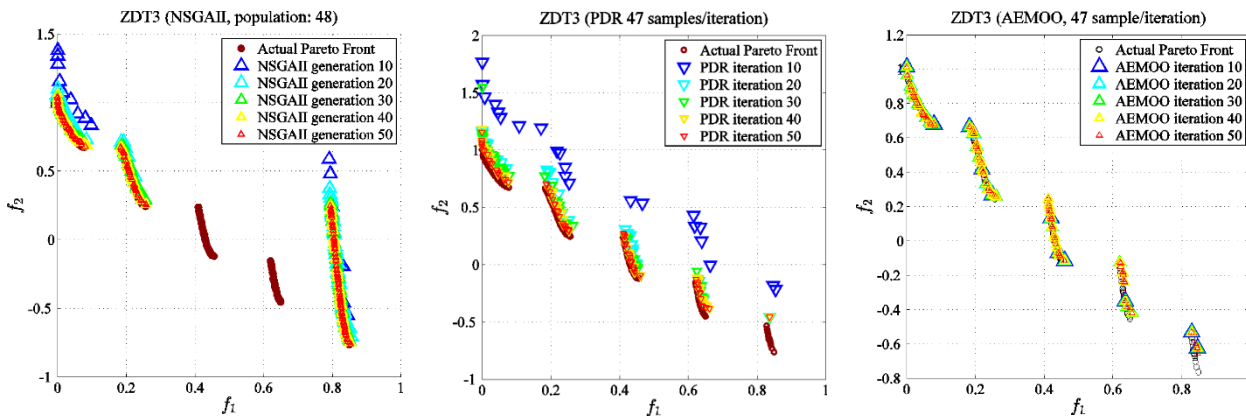


Figure 14: Example 4. Direct NSGAI (left), PDR (center) and metamodel-assisted AEMOO (right).

Summary and Future Work

A new methodology of approximating constraint boundaries in design, based on statistical classification, was implanted in LS-OPT and was applied to problems with binary, discontinuous and noisy functions. The classification algorithm implemented in LS-OPT is referred to as SVM, which is capable of defining highly nonlinear decision boundaries and disjoint regions, as seen in some of the examples. The SVM-based approach has a distinct advantage in my problems that have non-smooth responses. Additionally, using this approach one can construct a single boundary for multiple failure constraints. This is a very useful feature when system failure probability is calculated, as it allows a classifier to inherently include all the constraint dependencies while being constructed. In addition to non-smooth problems, the application of classifiers to adaptive sampling was also demonstrated, which again is a very useful feature that can increase the efficiency by avoiding sampling of unwanted regions of the design space.

Apart from the examples presented in the above section there can be several other applications of classifiers in design. Examples of binary response can be found in many disciplines and situations, e.g. in biomedical engineering, aeroelasticity, failed evaluations etc. Additionally, classifiers can be used to reduce the simulation time in MDO problems where multiple failure modes and solvers are involved. The simulation time can also be reduced if classification information is extracted before the termination time, which is possible in certain cases. They have applications in the field of material parameter identification as well.

Future work in this area entails the development of some special features as ready to use packages, such as pre-defined adaptive sampling algorithms, reliability-based design optimization with SVM boundaries, efficient global optimization using SVM constraints (EGOSVM) etc. Additionally, probabilistic classification models will also be implemented, such as probabilistic SVM (PSVM) and Random Forest classifiers. Another important development involves the integration of statistical classification with the post-processing tool (Viewer) of LS-OPT to turn it into a powerful machine learning tool.

Acknowledgement

We are thankful to Dr. Nielen Stander, the principal developer of LS-OPT, for several interesting discussions during the implementation of classifiers in LS-OPT and for creating the original flexible software framework into which the new features are integrated.

References

- [1] Wilson, A. G., & Fronczyk, K. M. Bayesian reliability: Combining information. *Quality Engineering*, 29(1), 119-129, 2017.
- [2] Alpaydin, Ethem. *Introduction to machine learning*. MIT press, 2014
- [3] James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*. Vol. 112. New York: springer, 2013.
- [4] Wang, G. Gary, and S. Shan. "Review of Metamodeling Techniques for Product Design with Computation-intensive Processes." *Proceedings of the Canadian Engineering Education Association* (2011).
- [5] Stander, N., Roux, W.J., Basudhar, A., Eggleston, T., Craig, K.J. *LS-OPT User's Manual*, June 2014.
- [6] Jones, Donald R., Matthias Schonlau, and William J. Welch. "Efficient global optimization of expensive black-box functions." *Journal of Global optimization* 13.4 (1998): 455-492.
- [7] Knowles, J. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. Technical report TR-COMPSYSBIO-2004-01, September 2004.
- [8] Bichon, Barron J., et al. "Efficient global reliability analysis for nonlinear implicit performance functions." *AIAA journal* 46.10 (2008): 2459-2468.
- [9] Youn, Byeng D., and Pingfeng Wang. "Bayesian reliability-based design optimization using eigenvector dimension reduction (EDR) method." *Structural and Multidisciplinary Optimization* 36.2 (2008): 107-123.

- [10] Basudhar, Anirban, Samy Missoum, and Antonio Harrison Sanchez. "Limit state function identification using support vector machines for discontinuous responses and disjoint failure domains." *Probabilistic Engineering Mechanics* 23.1 (2008): 1-11.
- [11] Lee, Herbert KH, et al. "Optimization subject to hidden constraints via statistical emulation." *Pacific Journal of Optimization* 7.3 (2011): 467-478.
- [12] Missoum, Samy, Palaniappan Ramu, and Raphael T. Haftka. "A convex hull approach for the reliability-based design optimization of nonlinear transient dynamic problems." *Computer methods in applied mechanics and engineering* 196.29 (2007): 2895-2906.
- [13] Hurtado, Jorge E. "Filtered importance sampling with support vector margin: a powerful method for structural reliability analysis." *Structural Safety* 29.1 (2007): 2-15.
- [14] Basudhar, Anirban, et al. "Constrained efficient global optimization with support vector machines." *Structural and Multidisciplinary Optimization* 46.2 (2012): 201-221.
- [15] Drucker, Harris, S. Wu, and Vladimir N. Vapnik. "Support vector machines for spam categorization." *Neural Networks, IEEE Transactions on* 10.5 (1999): 1048-1054.
- [16] Jain, Anil K., Robert P. W. Duin, and Jianchang Mao. "Statistical pattern recognition: A review." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.1 (2000): 4-37.
- [17] Schuldt, Christian, Ivan Laptev, and Barbara Caputo. "Recognizing human actions: a local SVM approach." *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 3. IEEE, 2004.
- [18] Vapnik, V.N., and Vapnik, V. *Statistical learning theory*. Vol. 1. New York: Wiley, 1998.
- [19] Saravanan, K., and S. Sasithra. "Review on classification based on artificial neural networks." *International Journal of Ambient Systems and Applications (IJASA) Vol 2*, no. 4 (2014): 11-18.
- [20] Layman, Ron, Samy Missoum, and Jonathan Vande Geest. "Simulation and probabilistic failure prediction of grafts for aortic aneurysm." *Engineering Computations* 27, no. 1 (2010): 84-105.
- [21] Jain, Anil K. "Data clustering: 50 years beyond K-means." *Pattern recognition letters* 31, no. 8 (2010): 651-666.
- [22] Duda, Richard O., Peter E. Hart, and David G. Stork. *Pattern classification*. John Wiley & Sons, 2012.
- [23] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).
- [24] Tay, Francis EH, and Lijuan Cao. "Application of support vector machines in financial time series forecasting." *Omega* 29, no. 4 (2001): 309-317.
- [25] Drucker, H., Wu, D. and Vapnik, V.N., 1999. Support vector machines for spam categorization. *IEEE Transactions on Neural networks*, 10(5), pp.1048-1054.
- [26] Díaz-Uriarte, Ramón, and Sara Alvarez De Andres. "Gene selection and classification of microarray data using random forest." *BMC bioinformatics* 7, no. 1 (2006): 3.
- [27] Keller, James M., Michael R. Gray, and James A. Givens. "A fuzzy k-nearest neighbor algorithm." *IEEE transactions on systems, man, and cybernetics* 4 (1985): 580-585.
- [28] Breiman, Leo. "Random forests." *Machine learning* 45, no. 1 (2001): 5-32.
- [29] Kohonen, Teuvo, Gyorgy Barna, and Ronald Chrisley. "Statistical pattern recognition with neural networks: Benchmarking studies." In *IEEE International Conference on Neural Networks*, vol. 1, pp. 61-68. 1988.
- [30] Anand, Rangachari, Kishan G. Mehrotra, Chilukuri K. Mohan, and Sanjay Ranka. "An improved algorithm for neural network classification of imbalanced training sets." *IEEE Transactions on Neural Networks* 4, no. 6 (1993): 962-969.
- [31] Basudhar, Anirban, and Samy Missoum. "A sampling-based approach for probabilistic design with random fields." *Computer Methods in Applied Mechanics and Engineering* 198.47 (2009): 3647-3655.
- [32] www.ncac.gwu.edu/vml/models.html, National Crash Analysis Center. Finite element model archive, 2008.
- [31] Basudhar, Anirban, and Samy Missoum. "Adaptive explicit decision functions for probabilistic design and optimization using support vector machines." *Computers & Structures* 86.19 (2008): 1904-1917.
- [32] Basudhar, Anirban, and Samy Missoum. "An improved adaptive sampling scheme for the construction of explicit boundaries." *Structural and Multidisciplinary Optimization* 42.4 (2010): 517-529.
- [33] Bourinet, J-M., F. Deheeger, and M. Lemaire. "Assessing small failure probabilities by combined subset simulation and support vector machines." *Structural Safety* 33.6 (2011): 343-353.
- [34] Basudhar, A. Multi-objective Optimization Using Adaptive Explicit Non-Dominated Region Sampling. *11th World Congress on Structural and Multidisciplinary Optimization*, Sydney, Australia, 2015.
- [35] Deb, Kalyanmoy, et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II." *Evolutionary Computation, IEEE Transactions on* 6.2 (2002): 182-197.
- [36] Stander, N. An Adaptive Surrogate-Assisted Strategy for Multi-Objective Optimization. *10th World Congress on Structural and Multidisciplinary Optimization*, Orlando, Florida, USA, 2013.