Adaptive Sampling using Support Vector Machines

D. Mandelli, C. Smith*

Idaho National Laboratory

**diego.mandelli@inl.gov*

INTRODUCTION

Reliability/safety analysis of stochastic dynamic systems (e.g., nuclear power plants, airplanes, chemical plants) is currently performed through a combination of Event-Trees and Fault-Trees [1]. However, these methods are characterized by the following disadvantages:

- Timing of events is not explicitly modeled
- Ordering of events is preset by the analyst
- The modeling of complex accident scenarios can be driven by expert judgment

For these reasons, there is currently an increasing interest in the development of dynamic PRA methodologies [2, 3, 4, 5] since they can be used to address the deficiencies of the conventional methods listed above.

However, while dynamic methodologies have distinct advantages over conventional methods, there is no general agreement about the need for dynamic methods due to the computational challenges. Computational challenges arise from the need to run many simulations in order to adequately propagate uncertainties and evaluate their impact in the analysis. Currently, state-of-practice for the analysis of dynamic stochastic systems and the propagation of uncertainties is performed using random sampling algorithms. This class of algorithms includes: Monte-Carlo [6, 7], stratified sampling (e.g., Latin Hypercube Sampling [8]), importance sampling [9] and orthogonal arrays based [10] algorithms.

Only recently, deterministic algorithms such as Polynomial Chaos Expansions [11] and Quasi Monte-Carlo [12] methods have started to be implemented. However, both random and deterministic sampling algorithms do not possess a sampling strategy that takes explicitly into account the results of previous simulations. On the other side, adaptive sampling algorithms perform a sampling strategy that chooses the next sample based on the results obtained by previous samples through a statistical learning models and, thus, focus sampling in risk sensitive regions such as boundaries between system safe and system failure: the limit surface.

The scope of this paper is to present a methodology based on Principal Component Analysis [13] and Support Vector Machines (SVMs) [14] for the determination of the limit surface. A set of simple test cases will be shown in order to show the implementation of the methodology.

ADAPTIVE SAMPLING

Propagation of uncertainties in complex systems such as nuclear power plants is usually performed by sampling algorithms which perform a series of simulation runs given a set uncertainty parameters [15]. Typically two problems arise at this point:

- The set of uncertain parameters is very large
- The computational costs are very high

Therefore, the space of the possible solutions, i.e., the issue space (each dimension corresponds to an uncertainty parameter), can be sampled only very sparsely and this precludes the ability to fully analyze impact of uncertainties on the system. The main idea behind algorithms such as Monte-Carlo or Latin Hypercube Sampling is to sample the issue space as uniformly as possible but the problems listed above still remain.

When sampling is applied for safety analysis applications, the following points often emerge:

- Set of parameters that are really of safety concern is a small subset of the original set of uncertainty parameters
- Many regions of the issue space are not of interest

The scope of adaptive sampling is to iteratively guide the choice of the next sample by analyzing the previous sampling history. Typically this is performed by building a surrogate model from the set of previous simulation runs and predicting the system behavior. In more detail three steps are needed:

1. Perform a set of initial simulation runs, i.e., training points; typically this is performed using classical sampling algorithms such as Monte-Carlo, Latin Hypercube Sampling, Voronoi Tessellation [16]
2. Build a surrogate model using the sampling results given in Step 1; Support Vector Machines, Regression based or Density based algorithms are typically chosen
3. Using the model built in Step 2, a set of candidate sample points are chosen
4. An importance parameter is assigned to each point chosen in Step 3 and the point with highest importance is chosen as next sample

A scheme of an adaptive sampling algorithm is given in Fig. 1.

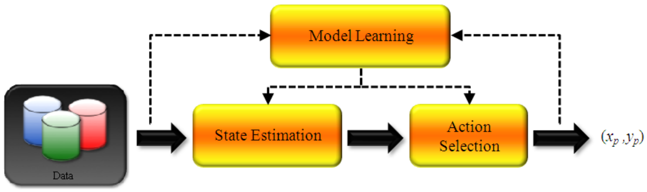

Fig. 1. Typical scheme of adaptive sampling algorithms

SUPPORT VECTOR MACHINES

Given a set of N samples $\mathbf{x}_i$ and their associated results $y_i = \pm 1$ (e.g., $y_i = +1$ for system success and $y_i = -1$ for system failure), the Support Vector Machine finds the boundary (i.e., the decision function) that separates the set of points having different y_i. The decision function lies between the support hyper-planes which are required to:

- Pass through at least one sample of each class (called support vectors)
- Not contain samples within them

For the linear case, see Fig. 2, the decision function is chosen such that distance between the support hyper-planes is maximized. They can be determined by solving the following system of equations:

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x} + b &= +1 \\ \mathbf{w} \cdot \mathbf{x} + b &= -1 \end{aligned} \quad (1)$$

Without going into the mathematical details, the determination of the hyper-planes is performed recursively and updated every time a new sample has been generated. Figure 2 shows the SVM decision function and the hyper-planes for a set of points in a 2-dimensional space having two different outcomes: $y_i = -1$ (green) and $y_i = +1$ (red) .

The transition from a linear to a generic non-linear hyperplane is performed using the the *kernel trick*. This process involves the projection of the original samples into a higher dimensional space known as featured space generated by kernel functions $K(\mathbf{x}_i, \mathbf{x}_j)$:

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

ALGORITHM

The methodology used in this paper can be summarized in the following three steps:

1. Generate an initial set of samples
2. Perform dimensionality reduction of the issue space (i.e., the number of uncertain parameters that really affect system dynamics) using dimensionality reduction algorithm

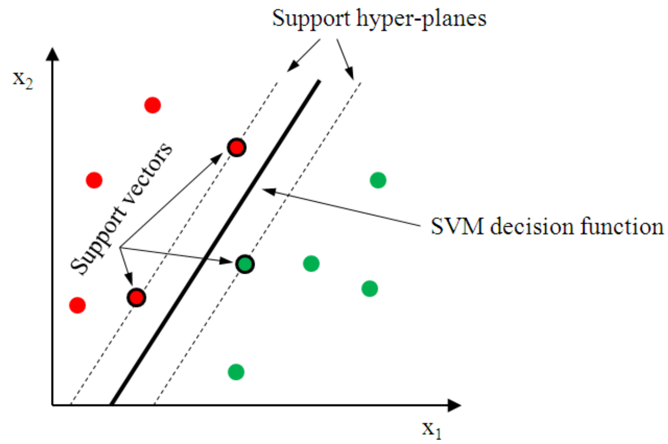

Fig. 2. Linear Support Vector Machines in a 2-dimensional space

3. Perform Support Vector Machine based sampling and evaluate decision function

For the scope of this paper we generated the initial set of samples using Latin Hypercube Sampling algorithm and the dimensionality reduction of the issue space using Principal Component Analysis [13]. In more detail, the algorithms for the Support Vector Machine based sampling is described as follows:

Algorithm 1 Adaptive Sampling Algorithm

1: Build the support decision function
2: Choose sample point on support decision function
3: Build the support decision function
4: Choose sample point on support hyper-plane and as far as possible from the one chosen in Step 2
5: Return to Step 1

TEST CASE

For the scope of this paper we will show an example using a 2-dimensional test case for the algorithm presented above. The limit surface has an arc shape and system success occurs below that arc and system failure above it. We applied the algorithm for such system and the results are shown in Fig. 3.

Figure 3 indicates how the density of the sample is very high near the boundaries and how the decision function approaches the limit surface.

CONCLUSIONS

In this paper we have shown a methodology that aims to find the limit surface for safety applications. This methodology uses a combination of dimensionality reduction and Support Vector Machine based sampling algorithms. Dimensionality has been performed on the space generated by the set of uncertainty parameters in order to reduce such number of parameters and thus focus on the most relevant ones. Determination of sampling strategy and evaluation of the limit surface has been performed using Support Vector Machine algorithms which gives flexibility for limit surface having non linear behavior. The simple test case presented in this paper demonstrates a graphical representation of the overall methodology.

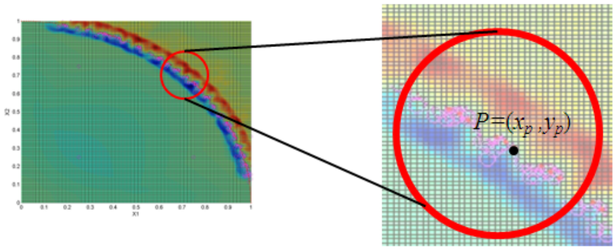

Fig. 3. Limit surface obtained for the test case

REFERENCES

1. E. ZIO, "Reliability engineering: Old problems and new challenges," *Reliability Engineering and System Safety*, **94**, 2, 125–141 (2009).
2. N. SIU, "Risk assessment for dynamic systems: an overview," *Reliability Engineering and System Safety*, **43**, 1, 43–73 (1994).
3. A. HAKOBYAN, T. ALDEMIR, R. DENNING, S. DUNAGAN, D. KUNSMAN, B. RUTT, and U. CATALYUREK, "Dynamic generation of accident progression event trees," *Nuclear Engineering and Design*, **238**, 12, 3457 – 3467 (2008).
4. T. ALDEMIR, M. STOVSKY, J. KIRSCHENBAUM, D. MANDELLI, P. BUCCI, L. MANGAN, D. MILLER, X. SUN, E. EKICI, S. GUARRO, M. YAU, B. JOHNSON, C. ELKS, and S. ARNDT, *NUREG/CR-6942: Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments*, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, Washington, DC (2007).
5. P. E. LABEAU, C. SMIDTS, and S. SWAMINATHAN, "Dynamic reliability: towards an integrated platform for probabilistic risk assessment. Reliability Engineering and Systems Safety," *Reliability Engineering and System Safety*, **68**, 219–254 (2000).
6. W. HASTINGS, "Monte Carlo samping methods using Markov chains and their applications," *Biometrika*, pp. 97–109 (1970).
7. J. S. LIU, *Monte Carlo strategies in scientific computing*, Springer (2001).
8. J. C. HELTON and F. J. DAVIS, "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems," *Reliability Engineering & System Safety*, **81**, 1 (2003).
9. J. GEWEKE, "Bayesian Inference in Econometric Models Using Monte Carlo Integration," *Econometrica*, **57**, 6, 1317–39 (November 1989).
10. OWEN, "Orthogonal arrays for computer experiments, integration and visualization," (1992).
11. H. N. NAJM, "Uncertainty Quantification and Polynomial Chaos Techniques in Computational Fluid Dynamics," *Annual Review of Fluid Mechanics*, **41**, 35–52 (Jan. 2009).
12. R. E. CAFLISCH, "Monte Carlo and quasi-Monte Carlo methods," *Acta Numerica*, **7**, 1–49 (1998).
13. I. T. JOLLIFFE, *Principal Component Analysis*, Springer, second ed. (October 2002).
14. C. J. C. BURGES, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Min. Knowl. Discov.*, **2**, 2, 121–167 (Jun. 1998).
15. E. Z. M. MARSEGUERRA, J. DEVOOGHT, and P. LABEAU, "A concept paper on dynamic reliability via Monte Carlo simulation," in "Mathematics and Computers in Simulation," (1998), pp. 47–371.
16. Q. DU, V. FABER, and M. GUNZBURGER, "Centroidal Voronoi Tessellations: Applications and Algorithms," *SIAM Rev.*, **41**, 4, 637–676 (Dec. 1999).