Applying Support Vector Machines to Uncertainty Quantification

Alexander Koran

McGill University

UNCERTAINTY QUANTIFICATION

Electrical components and systems are essential to modern society. From the microchips driving smartphones to large electrical transmission lines, these systems provide the technical basis for how the world operates.

As engineers, we aim to develop techniques to simulate and predict the output of these systems. Such techniques provide approximations of the actual systems and optimizing them can help increase confidence in the simulation results. This is the subject of the field of study called **Uncertainty Quantification.**

The Monte Carlo method is frequently used to simulate electrical circuits. This involves repeated selection of input parameter values, computation of output values, and generation of an output probability density curve. Unfortunately, many modern systems have such a high complexity and number of uncertain parameters that the computation time becomes excessive.

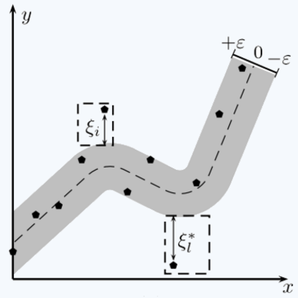
The goal of this project was to study and find alternative simulation techniques with lower computation time and power consumption.

SVM and LS-SVM

One approach is to apply Machine Learning techniques to the area of Uncertainty Quantification. More specifically, **Support Vector Machines** (SVMs) can be used to predict the output distribution.

In this work, we use a Support Vector Machine to identify the best hyperplane to separate and classify data. SVMs can also be used for regression, in a sense finding the best fitting hyperplane. Since the data is not always linearly separable, we may transform it into a form preferable for a regression fit using a Kernel function.

SVMs are a useful choice of model when accuracy is preferred over extreme precision. The loss function within the SVM only penalizes sample points outside the “epsilon tube”, that is, more than one epsilon distance away from the prediction function.



The **Least Squares SVM** is a variant of SVM. Using the errors between the predicted and actual values, it follows a scheme similar to the least squares regression to find the best fit.

CALCULATIONS

Due to complexity, some derivation details are not included here. We would like our prediction SVM function to have the following format:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \beta_i K(\mathbf{x}, \mathbf{x}_i) + b$$

where K is our chosen Kernel function, x_i is our training sample input, and $\boldsymbol{\beta}_i$ and b are parameters that need to be computed. Using all the SVM constraints, the following optimization problem should be solved:

$$\begin{cases} \min_{\mathbf{w},b,\xi^*,\xi} & \frac{1}{2}\mathbf{w}'\mathbf{w} + \frac{1}{2}b^2 + \frac{C}{2}\left[\sum_{i=1}^{n} w_p \xi_i^2 + \sum_{i=1}^{n} w_n \left(\xi_i^*\right)^2\right] \\ \text{s.t.} & y_i - \mathbf{w}'\phi(\mathbf{x}_i) - b \le \epsilon + \xi_i \quad \text{and} \\ & \mathbf{w}'\phi(\mathbf{x}_i) + b - y_i \le \epsilon + \xi_i^*, \ \text{for } i = 1, 2, \ldots, n \end{cases}$$

After substitutions and algebraic manipulations, we can solve this optimization problem with Lagrange Multipliers and KKT conditions i.e. by setting the gradient equal to zero. Using matrix calculus, we arrive at the following compact equation:

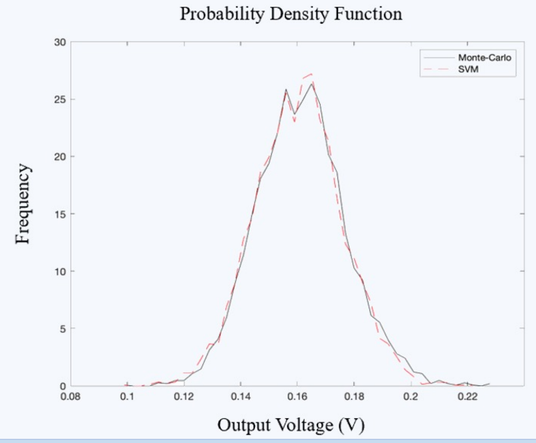
$$\begin{pmatrix} \frac{\mathbf{I}}{C} + \mathbf{DK} & \mathbf{D1} \\ \mathbf{1}'\mathbf{DK} & \mathbf{1}'\mathbf{D1} + \frac{1}{C} \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta} \\ b \end{pmatrix} = \begin{pmatrix} \mathbf{D}(\mathbf{y} - \mathbf{e}) \\ \mathbf{1}'\mathbf{D}(\mathbf{y} - \mathbf{e}) \end{pmatrix} = \begin{pmatrix} \mathbf{I} \\ \mathbf{1}' \end{pmatrix} \mathbf{D}(\mathbf{y} - \mathbf{e})$$

This system can be solved not directly, but by using an iterative procedure.

RESULTS

Both the SVM and LS-SVM models were implemented in MATLAB. The Probability Density Functions (PDF) of the SVM model (primary focus) and the Monte-Carlo method are compared below.

The SVM model does a very similar job to MC, but with significantly less computational power and sample points. In this case, there were 400 training samples, 5 uncertain parameters, and 5000 testing samples.

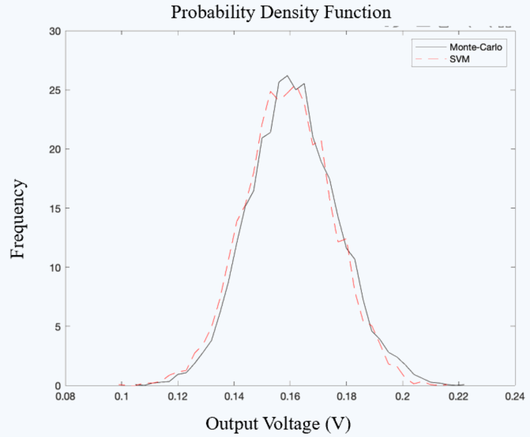


REDUCTION OF TRAINING DATA

The field of Uncertainty Quantification focuses on developing accurate predictions without large quantities of input data. Experiments were designed to determine how few samples we can train on to still get accurate results.

We started with 50 training samples and increments of 10 for the number of samples in each iteration. A Validation Set was used to calculate the mean squared error between the predicted and actual values. Once this error dropped beneath a tolerance, we stopped the iterations and used this number of training samples. Finally, we tested this model on larger test sets.

Using an error tolerance of 1e-5, the iterations stopped at 120 samples. Shown below is the resulting PDF predicted from the SVM compared with the Monte-Carlo generated PDF.



CONCLUSION

The application of SVMs was used and subsequently refined to simulate a specific circuit with far fewer training samples than classic Monte Carlo simulation approaches. A refined SVM was developed that enables accurate predictions using only 120 samples.

FUTURE WORKS

Improve the SVM model through better hyperparameter tuning methods, such as Simulated Annealing and Optimization techniques for initial values.

Further reduce the number of training samples needed by using different SVM variants and improving sampling techniques.

ACKNOWLEDGEMENTS

I would like to thank Professor Khazaka, Karan Sidhu, and William Wang for their continued guidance and support of this project.