



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Constrained Classification for Infrastructure Threat Assessment

K. P. Lennox, L. Glascoe

June 6, 2011

IEEE Conference on Technologies for Homeland Security
Waltham, MA, United States
November 15, 2011 through November 17, 2011

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Constrained Classification for Infrastructure Threat Assessment

Kristin P. Lennox and Lee Glascoe
Lawrence Livermore National Laboratory
Livermore, CA USA
E-mail: lennox3@llnl.gov

Abstract—Validated computer simulation is an important aspect of critical infrastructure vulnerability assessment. The high computational cost of such models limits the number of threat scenarios that may be directly evaluated, which leads to a need for statistical emulation to predict outcomes for additional scenarios. Our particular area of interest is statistical methods for emulating complex computer codes that predict if a particular tunnel/explosive configuration results in the breaching of an underground transportation tunnel. In this case, there is considerable *a priori* information as to the properties of this breach classification boundary. We propose a constrained classifier, in the form of a parametric support vector machine, that allows us to incorporate expert knowledge into the shape of the decision boundary. We demonstrate the effectiveness of this technique with both a simulation study and by applying the method to a tunnel breach data set. This analysis reveals that constrained classification can offer substantial benefits for small sample sizes. The technique may be used either to provide a final classification result in the face of extremely limited data or as an interim step to guide adaptive sampling.

Index Terms—support vector machines, computer experiments, statistical learning

I. INTRODUCTION

Validated high fidelity computations are increasingly employed to supplement costly physical experiments; further, such modeling, when carefully used, can estimate structural response of systems that cannot be feasibly tested. As computer codes increase in capability and utility, computational cost typically increases. Therefore, when a large number of input configurations need to be modeled, this is often accomplished through the use of a statistical “emulator” that uses a small set of expensive code runs to characterize the complete input space [1], [2]. For example, the use of statistical learning methods has been examined to approximate limit state functions in reliability analysis [3].

Infrastructure threat assessment is a field that can particularly benefit from the application of computer modeling and corresponding emulation. Consider, for example, the assessment of explosive threats to tunnel systems [4]. The goal is to determine what combinations of explosive threats and tunnel geometries could lead to a tunnel “breach.” Here we define a

tunnel breach simply as damage which would allow material from outside of the tunnel to flow or fall into the tunnel. The input space of interest consists of the wall thickness of the tunnel, the mass of the explosive charge, and the distance from the surface of the charge to the tunnel wall (standoff distance).

For any given tunnel-charge configuration, simulations can be used to predict the effect of the blast and classify the tunnel as either breached or not. The full three-dimensional physics model is very costly to evaluate, so only a relatively small number of simulation runs will be available to characterize the full conformation space.

Since full simulation runs can be performed for only a limited number of input conformations, they should be chosen so as to maximize the information generated about the breach/no breach boundary. Even without any simulation runs, certain characteristics of the breach boundary are known: the breach region should be contiguous, the risk of breach should increase as charge size increases, decrease as standoff increases, and decrease as wall thickness increases. Incorporating this information into the sampling scheme should reduce the number of data points required to completely characterize the breach curve.

This is not a typical classification problem. In general, sophisticated statistical learning algorithms (e.g. support vector machines, random forests, and neural networks) are used for nonparametric classification: minimal assumptions are made about the characteristics of the underlying data or the classification surface itself. In contrast, parametric classification models, such as quadratic discriminant analysis, make assumptions about the underlying probabilistic nature of the classification features, which is also inappropriate in this case. The breach/no breach data is frequently separable for small to moderate sample sizes, which limits the utility of logistic regression.

We therefore propose a version of the support vector machine, the parametric SVM, that has the ability to incorporate expert knowledge into the classification boundary. By sacrificing some of the flexibility of the standard SVM formulation, we gain the ability to limit the boundary shape to an explicitly defined class of functions. Parametric SVMs also have interpretable model parameters which are functions of support vectors and Lagrange multipliers. In particular, an intercept value is directly related to the standard SVM bias term, which allows constraints to be placed. Note that

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This work was sponsored by the Department of Homeland Security, Science and Technology Directorate, Infrastructure Protection and Disaster Management Division.

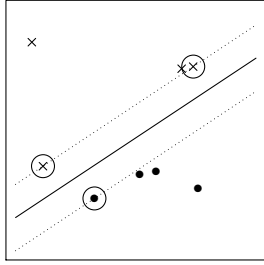


Fig. 1. A maximal margin hyperplane, in this case a line, between two classes. Dotted lines represent the margin, while circled points are support vectors.

this form of constrained classification is different from the alternative use of the term that refers to placing constraints on error rates [5]. Section II gives an overview of classification with support vector machines. In Section III we present the parametric SVM, and discuss how constraints can be placed upon a decision boundary and the bias parameter. Section IV contains a simulation study of both parametric and intercept constrained parametric support vector machines under adaptive sampling, and Section V examines the feasibility of applying this methodology to tunnel simulation data. In Section VI we discuss our conclusions.

II. SUPPORT VECTOR MACHINES OVERVIEW

The support vector machine (SVM) is a method for generating optimal decision surfaces separating two classes [6]. Consider a set of observations consisting of $\mathbf{z}_1, \dots, \mathbf{z}_n$, where $\mathbf{z}_i = (z_{i0}, \dots, z_{im})^T$. Each observation has a class label $d_i \in \{-1, 1\}$. The goal is to identify a decision rule that can predict the class label of a new observation \mathbf{z}_{n+1} based on the existing labeled data. Assuming that the data is linearly separable, there are infinitely many hyperplanes of the form

$$0 = \mathbf{a}^T \mathbf{z} + \rho \quad (1)$$

which will perfectly separate the existing data. To ensure a unique solution, we can specifically consider the maximal margin hyperplane: the hyperplane of the form given in (1) which perfectly separates the data and for which the minimum distance to an observed point, or margin, is maximized. In particular, this is accomplished by minimizing $\|\mathbf{a}\|^2$ over \mathbf{a}, ρ subject to the constraint $d_i(\mathbf{a}^T \mathbf{z}_i + \rho) \geq 1$. An example of such a boundary is shown in Fig. 1. Note how the margin, and therefore the decision boundary, is defined by a small subset of points.

The support vector machine generalizes this concept in two important ways: allowance for nonlinear boundaries and misclassified points. The final classification function is defined in terms of the existing data and Lagrange multipliers as follows:

$$g(\mathbf{z}) = \rho + \sum_{i=1}^n \lambda_i d_i K(\mathbf{z}_i, \mathbf{z}). \quad (2)$$

Here K is a kernel function that may map the data into a higher, and possibly infinite, dimensional space, and the

margin maximization step includes “slack variables” and a cost parameter. In brief, slack variables and the cost parameter allow some points to fall within the margin or even be misclassified. As the cost parameter increases, the penalty for a point falling inside the margin increases as well. A “soft-margin” SVM, which allows misclassification, increasingly resembles a “hard-margin” SVM, which does not, as the cost goes to infinity. Common kernel functions include the linear kernel ($K(\mathbf{z}_1, \mathbf{z}_2) = \mathbf{z}_1^T \mathbf{z}_2$), radial basis kernel ($K(\mathbf{z}_1, \mathbf{z}_2) = \exp(-\gamma \|\mathbf{z}_1 - \mathbf{z}_2\|^2)$), and the d th-degree polynomial kernel ($K(\mathbf{z}_1, \mathbf{z}_2) = (1 + \mathbf{z}_1^T \mathbf{z}_2)^d$) [7]. A detailed discussion of the role of kernel functions, Lagrange multipliers, and slack variables is beyond the scope of this paper, but may be found in many introductions to SVMs and statistical learning methodologies, including [7]–[9].

The sign of $g(\mathbf{z})$ gives the classification of the new point. Note that, in general, most of the coefficients λ_i will be equal to 0, meaning that only a small subset of the data will need to be retained in order to classify new points. These points are known as support vectors.

III. CONSTRAINED SUPPORT VECTOR MACHINES

The support vector machine is generally used in a non-parametric classification context. The flexibility offered by the high-dimensional mapping generated by standard kernel functions allows the separating surface to take on a tremendous variety of shapes, including noncontiguous ones. However, in practical applications this flexibility may be unnecessary, and may in fact hinder effective classification.

Of particular interest is the scenario in which the classification boundary may be defined as a function of m of the classification features. Without loss of generality, we may say that this excluded feature corresponds to z_{i0} , and further rename it y_i . We will also henceforward refer to our “explanatory” variables (z_1, \dots, z_m) as $\mathbf{x} = (x_1, \dots, x_m)$. In order for the decision boundary to meet the definition of a function, $g(y, \mathbf{x}) = 0$ and $g(y', \mathbf{x}) = 0 \Rightarrow y = y'$ for each \mathbf{x} . This allows a function $f(\mathbf{x}) = (y|g(y, \mathbf{x}) = 0)$ to be defined. If it may further be assumed that $\text{sign}(y_{n+1} - f(\mathbf{x}_{n+1})) = (-1)^k \text{sign}(g(y_{n+1}, \mathbf{x}_{n+1}))$, where k is fixed as either 0 or 1, then the SVM may be considered to define a “critical surface” in y , such that for any \mathbf{x} there exists a value for y above which observations receive one classification and below which they receive another.

It is possible to generate such boundary functions using SVMs with a linear kernel. The fact that data is separated in a known space, rather than the potentially high dimensional space implied by other kernels, allows for a unique degree of interpretability for the coefficients in the classification function. In particular, we can define $f(\mathbf{x})$ as:

$$f(\mathbf{x}) = b_0 + \sum_{j=1}^m b_j x_j \quad (3)$$

where

$$b_j = \frac{-(\sum_{i=1}^n \lambda_i d_i x_{ji})}{\sum_{i=1}^n \lambda_i d_i y_i} \quad (4)$$

and

$$b_0 = \frac{-\rho}{\sum_{i=1}^n \lambda_i d_i y_i}. \quad (5)$$

given $\sum_{i=1}^n \lambda_i d_i y_i \neq 0$. Note that when this assumption is violated, the SVM is ignoring y for classification purposes, and so defining a classification surface in terms of y is not reasonable. The classification predicted for a new observation (y, \mathbf{x}) is given by:

$$\begin{aligned} \text{sign}(y - f(\mathbf{x})) & \quad \text{if } \sum_{i=1}^n \lambda_i d_i y_i < 0 \\ -\text{sign}(y - f(\mathbf{x})) & \quad \text{if } \sum_{i=1}^n \lambda_i d_i y_i > 0 \end{aligned}$$

Note that this formulation gives us coefficients that correspond to particular variables, allowing us to determine their relationships and relative importance. This is a contrast to the standard SVM formulation in which the coefficients λ_i give information about the impact of a particular observation, rather than a particular variable.

This formulation is not limited to modeling linear decision surfaces. A fixed transformation into a higher dimensional feature space may be performed if desired. In order to fit a more complex classification surface, functional terms $\{p_j(\mathbf{x})\}_{j=1}^k$ may be included and the linear SVM fit in the $k + 1$ dimensional space defined by $(y, p_1(\mathbf{x}), \dots, p_k(\mathbf{x}))$.

For example, consider data of the form (y_i, x_{i1}, x_{i2}) . A quadratic separating surface can be generated by fitting a linear SVM on the feature vector $(y_i, x_{i1}, x_{i1}^2, x_{i2}, x_{i2}^2, x_{i1}x_{i2})$. The resulting decision boundary may be written as:

$$f(x_1, x_2) = b_0 + b_1 x_1 + b_2 x_1^2 + b_3 x_2 + b_4 x_2^2 + b_5 x_1 x_2. \quad (6)$$

This strategy is not equivalent to using an SVM with a polynomial kernel of degree two, for which the optimal decision surface may not be a function with response y . That is, there may exist a vector \mathbf{x} and values $y \neq y'$ such that $g(y, \mathbf{x}) = g(y', \mathbf{x}) = 0$.

The functions $p_j(\mathbf{x})$ need not necessarily be terms in a polynomial expansion, and are at the discretion of the user.

Interpretability of the coefficients b_0, \dots, b_k also allows meaningful constraints to be placed at optimization time. Specifically we will explore the imposition of equality constraints on the intercept term b_0 .

If the component functions of the parametric SVM all satisfy the constraint $p_j(\mathbf{0}) = 0$, where $\mathbf{0}$ is the 0 vector, then b_0 is the y -intercept for the classification surface. Recall that the standard SVM formulation is not defined explicitly in terms of b_0 . Rather it is defined in terms of the Lagrange multipliers $\{\lambda_i\}_{i=1}^n$ and the bias term ρ as shown in (5). The parametric SVM context is already restricted such that $\sum_{i=1}^n \lambda_i d_i y_i \neq 0$, so the intercept will always be well defined. Therefore, the equality constraint $\rho = 0$ is equivalent to the intercept constraint $b_0 = 0$. Note that recentering the data will allow a constraint of this form to force the parametric SVM boundary to pass through any one point in the input space.

IV. ADAPTIVE SAMPLING STUDY

A simulation study was performed to compare the performance of three classifiers: a nonparametric SVM with a radial basis kernel, a parametric SVM trained on data of the form (y, x, x^2) , and the same parametric SVM constrained to have $b_0 = 0$. Two simulated boundaries in two-dimensional space were considered: $y = -x^2 + 2x$ and $y = e^x - 1$. In both cases the assumptions about the classification boundary being expressible as a function of x and the intercept hold, however only one case has the exact functional form of the parametric SVM classifier. The purpose of the study was to evaluate the classifiers for both small fixed sample sizes and under adaptive sampling.

Adaptive sampling is a technique used to minimize the number of data points needed to produce an accurate final model by querying new points based on a model fit to existing data. The basic premise is that points should be sampled in regions where there is the greatest potential for model improvement [10].

For the simulation study, an initial space filling sample of size 10 was chosen using Latin hypercube sampling [11]. In the SVM framework, points far from the classification boundary have no impact on the classifier, and conversely points on the boundary itself have the greatest influence. Existing strategies for adaptive sampling with SVMs therefore focus on points on or near boundary [12]–[14]. Therefore a SVM was trained on the existing data, and a large set of candidate points located on the decision boundary was generated. The point with the maximum minimum distance to an existing point was selected for addition to the training set. The process was then repeated on the remaining candidate points, each time including all previously selected points in the maximin evaluation step, until five new points had been chosen. All points in the newly selected set were evaluated, and the classifier retrained. This strategy is similar to that of [13] in that it ensures that successive selected points are located on the classification boundary, but are distant from existing sampled points and each other. Adaptive sampling was carried out for a total of four iterations yielding a final total of 30 data points.

The cost parameter C was permitted to take values between 2^{-5} and 2^{15} , while the γ parameter for the radial basis kernel could take values between 2^{-15} and 2^5 . When sufficient data points from both classes were observed, tuning was carried out using cross-validation. Otherwise default settings of $C = 100$ and $\gamma = 0.5$ were used. All three classification methods were trained on the same 100 LHS starting sets, allowing direct comparisons to be made for each initial data set. Calculations were carried out in R [15] using the e1071 [16] and Liblinear [17] packages.

The overall classification accuracy of each method was assessed on a grid of 10^6 points after each iteration. The median accuracies for the 100 replicates at each iteration are shown in Fig. 2. Notice that the differences in median accuracy between methods decrease dramatically over the

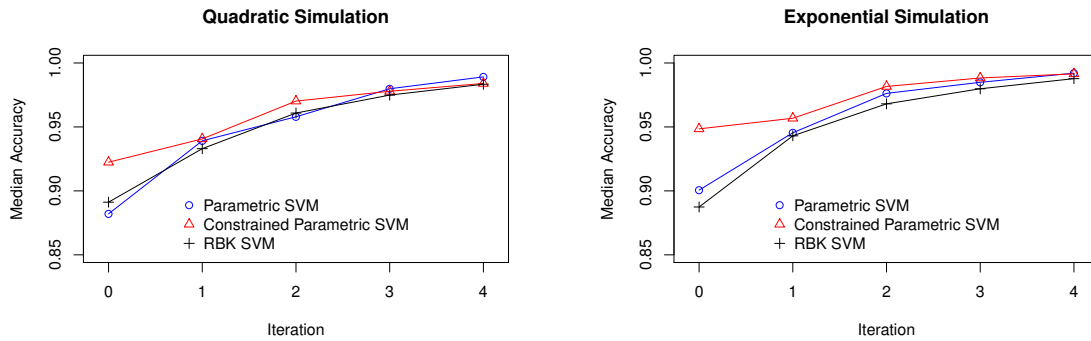


Fig. 2. Median accuracy across initial sampling and four iterations of adaptive sample for each of the three classification methods.

TABLE I

DIRECT COMPARISON OF CONSTRAINED AND UNCONSTRAINED SVMs. THE TABLE GIVES THE PROPORTION OF THE REPLICATES FOR WHICH THE PARAMETRIC SVMs HAD HIGHER GLOBAL ACCURACY THAN THE RBK SVM.

Quadratic Simulation		
Sample Size	10	30
Parametric SVM	0.45	0.74
Constrained Parametric SVM	0.72	0.58
Exponential Simulation		
Sample Size	10	30
Parametric SVM	0.61	0.69
Constrained Parametric SVM	0.90	0.68

four adaptive sampling iterations. Rather than considering the magnitude of the difference in accuracy, comparisons are made based on the proportion of simulation samples for which one method outperformed another. Table I gives the proportion of simulation runs for which the parametric or constrained parametric SVMs performed better than the unconstrained RBK SVM for the initial sample of 10 and the final sample of 30. Where p-values are mentioned in this analysis, they refer to the binomial test with the alternative hypothesis that the parametric SVM provides superior performance as compared to the RBK SVM more the half of the time.

For both simulation studies, the constrained parametric SVM is clearly outperforming the other methods for the initial data set, reinforcing our intuition that knowledge-based constraints add considerably to accuracy when data is limited. In both cases a binomial test of superiority of the constrained parametric SVM to the RBK SVM is highly significant (p-value $< 1e-5$). Fig. 3 illustrates the advantage of constrained versus unconstrained classification in this case. The nonparametric SVM places the classification boundary far from the true value, which both decreases accuracy for the current iteration and decreases the value of adaptive sampling. In contrast the constrained parametric SVM is very close to the true boundary, and any samples chosen for the next iteration will be of high value. For the exponential test case, this advantage is maintained throughout the adaptive sampling process, with a p-value of $2e-4$ for a binomial test of superiority for the complete samples. In contrast, for the

quadratic example the advantage dwindled to the point where it is no longer significant at the 0.05 level by the fourth adaptive sampling iteration.

For the initial sample, the unconstrained parametric SVM underperforms the RBK for the quadratic target, though not by a statistically significant amount. However, after four iterations of adaptive sampling this method outperforms the RBK SVM for both the quadratic and exponential simulations a significant proportion of the time (p-values less than $1e-5$ and $1e-4$ respectively). These results suggest that the constrained parametric SVM has a substantial advantage for small data sets, but benefits less from adaptive sampling than does the unconstrained parametric classifier. The version without an intercept constraint is not always superior for initial samples, but shows a considerable advantage over the RBK SVM after several iterations of adaptive sampling.

Surprisingly, the advantage of the parametric SVM models is most evident in the non-quadratic target case, with both quadratic-based models surpassing the RBK model during both initial and final modeling stages. This indicates that an exact match between the functional form of the SVM and the that of the target boundary is not necessary for an improvement over the fully nonparametric classifier.

In no case did either of the parametric methods perform significantly worse than the fully nonparametric model, and in most they were significantly better. As a caveat, these simulations involved neither substantial model misspecification nor large sample sizes. The combination of the two could cause a parametric SVM to considerably underperform an RBK model. When using the parametric SVM, we recommend that diagnostics be employed to identify potential poor model fits at each adaptive sampling iteration.

V. APPLICATION: TUNNEL BREACH CLASSIFICATION

Recall that our motivating example involves computer modeling to predict tunnel breach in response to an explosive threat. These simulations are computationally costly, so an algorithm that provides high accuracy with fewer samples would be of considerable value.

The test data set contains 377 breach simulations. As described previously, each observation consists of a threat conformation represented by charge mass and standoff, tunnel conformation represented by wall thickness, and a breach

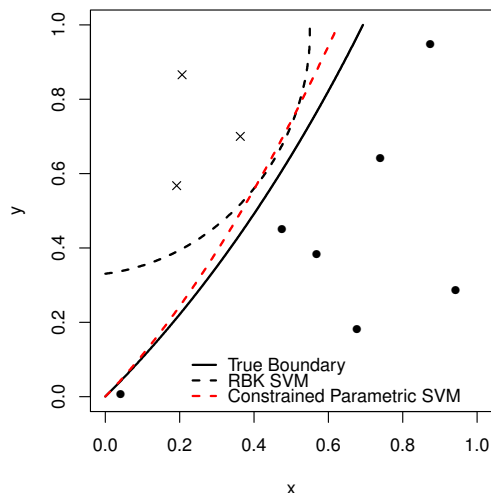


Fig. 3. Example of RBK SVM and constrained parametric SVM for sample of size 10 with an exponential classification boundary. The accuracy for the RBK SVM is 0.895 while the accuracy for the constrained parametric SVM is 0.964. The unconstrained parametric SVM for this data is not pictured, but has global accuracy of 0.927.

or no breach label. Due to the complex nature of a breach failure, there will be some noise near the breach boundary, so a soft-margin SVM is appropriate. The minimum charge weight for this data set constitutes a “threshold charge.” No contact charge with less mass is expected to cause breach for the thinnest tunnel wall studied. All data was rescaled to fall between 0 and 1 prior to either nonparametric SVM classification or fixed mapping into a higher dimensional space for parametric SVM classification.

Once again, three competing models were considered for this data: a standard SVM with a radial basis kernel, a parametric SVM without additional constraints, and a parametric SVM with the intercept constrained to be equal to 0. The parametric SVM treats the standoff distance as the response variable and the charge mass and wall thickness as explanatory variables. Thus this model identifies a “critical standoff” for each charge mass and wall thickness: a charge of the given size placed less than the critical distance from the wall will be predicted to cause a breach. Our initial belief about the shape of this classification surface is that it should be smoothly decreasing as wall thickness increases, smoothly increasing as charge mass increases, and not contain a large number of inflection points. The transformation chosen will give a classification surface of the form:

$$y = b_0 + b_1x_1 + b_2x_1^2 + b_3x_1^{0.5} + b_4x_2 + b_5x_2^2 + b_6x_2^{0.5} + b_7x_1x_2 \quad (7)$$

where y is the critical standoff, x_1 is charge mass and x_2 is wall thickness. This gives a relatively broad class of functions which will generally satisfy the previously mentioned restrictions even for fairly small amounts of data. In practice, the intercept constraint forces the classifier to indicate that there is no chance of breach for values less than our minimum threshold charge.

TABLE II

ACCURACY OF METHODS ON TUNNEL BREACH DATA. SUBSETS OF THE DATA WERE USED FOR TRAINING, AND THE REMAINING POINTS WERE SUBSEQUENTLY PREDICTED. FOR THE FULL DATA SET, LEAVE-ONE-OUT CROSS VALIDATION WAS USED TO ASSESS ACCURACY.

Training Set	50	100	200	300	Full Data
RBK SVM	0.670	0.765	0.684	0.714	0.836
PSVM	0.670	0.783	0.678	0.753	0.798
Const. PSVM	0.826	0.801	0.718	0.753	0.833

This data was generated using adaptive sampling, although the original classification method was not equivalent to any of the SVM formulations under consideration. Due to the nonrandom nature of the sampling, the observations are not of uniform predictive value: points which were added to the set later tend to be closer to the classification boundary, and therefore more difficult for a method to correctly classify. Conversely, they are more valuable to a classifier since they contain better information. The data set included the order in which points were chosen for evaluation, so that it was possible to train on the first n points sampled and test on the remaining points. All three methods were trained on the first 50, 100, 200, and 300 points, and tested on the remaining values. Leave-one-out cross validation was also used to examine the accuracy of the methods trained on the full data set. Parameter tuning was carried out as described in the simulation section. Results are given in Table II. Note that accuracy does not uniformly increase as training set size increases for any method, again due to the adaptive nature of the sampling and the relative difficulty of predicting breach classification for later points.

The benefits of using the intercept constrained parametric SVM are evident for the smallest sample case, with a substantial difference for the set of size 50. For this test case, McNemar’s test [18] comparing the accuracy of the intercept constrained SVM and the nonparametric SVM gives a p-value less than 10^{-6} , indicating that the difference is statistically significant. For the larger sample sizes, the constrained SVM generally performs better, but the differences are not statistically significant. The unconstrained parametric model was never significantly different from the nonparametric SVM. It appears that the primary benefit of the parametric SVM in this case comes from the constraint placed on the intercept term. Conversely, there is no evidence that anything is lost by using a parametric SVM for the sample sizes under consideration. This suggests that the potential problems caused by an overly constrained parametric model are not in evidence here.

Fig. 4 shows the constrained parametric versus nonparametric classification boundaries at a fixed wall thickness for the both the first 100 samples and the full data set. The classification boundaries for the full data set are very similar for the two methods. However, the nonparametric SVM boundary based on the first 100 data points is a poor match to the either of the full data boundaries, in addition to failing to respect the intercept constraint. In contrast, the constrained

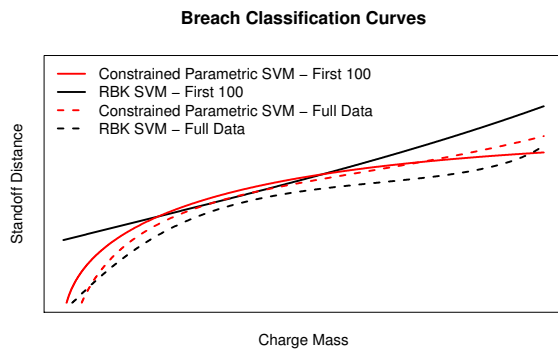


Fig. 4. Comparison of the constrained parametric SVM and nonparametric SVM for a fixed wall thickness value.

parametric SVM gives similar results for both the reduced and full training sets. This once again emphasizes the advantage of incorporating all available information into data classification efforts. The improvement in performance for this reduced sample size indicates that using a constrained parametric SVM may lead to the identification of a reliable breach boundary more quickly than the nonparametric alternative.

While strong conclusions should not be drawn based on a single nonrandom data set, these results indicate that a well-formulated parametric SVM, particularly one with an accurate intercept constraint, could be a valuable tool for characterizing future breach curves, particularly during the early stages of adaptive sampling.

VI. DISCUSSION

We have presented a variant of the support vector machine that is suited to the situation when there is *a priori* information about the nature of the boundary between classes. This method gives more interpretability to the model coefficients, which may now be explicitly associated with particular explanatory variables. In particular, under mild restrictions, the constraint $\rho = 0$ is equivalent to the y -intercept of the decision surface being equal to 0.

A simulation study showed the advantages of our parametric SVM formulation over a standard radial basis kernel SVM. The intercept constrained formulation performed particularly well for small initial sample sizes, while the unconstrained quadratic model performed best after successive applications of adaptive sampling. A mild mismatch between the choice of parametric formulation and the true functional form did not remove these advantages.

A data set examining the vulnerability of a hypothetical tunnel geometry to blast showed evidence that a constrained parametric SVM could provide an improvement over a non-parametric SVM for the purposes of modeling and adaptive

sampling. A constrained model that can rapidly refine the search space during the early stages of sampling would reduce the computational cost of developing reliable tunnel breach assessments. Once trained, such a classification method can be used to create a “fast-running” tool to examine potential tunnel vulnerabilities. This technique could also be applied to other classification problems where the decision boundary can be described in an explicit functional form.

REFERENCES

- [1] T. Santner, B. Williams, and W. Notz, *The Design and Analysis of Computer Experiments*. Springer Verlag, 2003.
- [2] K. Fang, R. Li, and A. Sudjianto, *Design and Modeling for Computer Experiments*. CRC Press, 2006.
- [3] J. Hurtado, “An examination of methods for approximating implicit limit state functions from the viewpoint of statistical learning theory,” *Structural Safety*, vol. 26, no. 3, pp. 271–293, 2004.
- [4] L. G. Glascoe, J. P. Morris, L. A. Glenn, and M. Krnjajic, “Development and application of a fast-running tool to characterize shock damage within tunnel structures,” in *Proceedings of the Fifth Congress ASCE Forensics Engineering*. ASCE, 2009, pp. 13–13. [Online]. Available: <http://link.aip.org/link/?ASC/362/13/1>
- [5] A. Bounsiar, P. Beausery, and E. Grall, “A straightforward SVM approach for classification with constraints,” in *Proc. EUSIPCO*, vol. 5, 2005.
- [6] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.
- [8] C. Bishop, *Pattern Recognition and Machine Learning*. Springer New York, 2006.
- [9] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2006.
- [10] D. Cohn, L. Atlas, and R. Ladner, “Improving generalization with active learning,” *Machine Learning*, vol. 15, no. 2, pp. 201–221, 1994.
- [11] M. McKay, R. Beckman, and W. Conover, “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.
- [12] G. Schohn and D. Cohn, “Less is more: Active learning with support vector machines,” in *Machine Learning International Workshop then Conference*, 2000, pp. 839–846.
- [13] A. Basudhar and S. Missoum, “Adaptive explicit decision functions for probabilistic design and optimization using support vector machines,” *Computers & Structures*, vol. 86, no. 19–20, pp. 1904–1917, 2008.
- [14] —, “An improved adaptive sampling scheme for the construction of explicit boundaries,” *Structural and Multidisciplinary Optimization*, pp. 1–13, 2010.
- [15] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2011, ISBN 3-900051-07-0. [Online]. Available: <http://www.R-project.org>
- [16] E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, and A. Weingessel, *e1071: Misc Functions of the Department of Statistics (e1071)*, TU Wien, 2011, R package version 1.5-25. [Online]. Available: <http://CRAN.R-project.org/package=e1071>
- [17] T. Helleputte, *LiblineaR: Linear Predictive Models Based On The Liblinear C/C++ Library*, 2011, R package version 1.80-4.
- [18] Q. McNemar, “Note on the sampling error of the difference between correlated proportions or percentages,” *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.