Optimal ν-SVM Parameter Estimation using Multi Objective Evolutionary Algorithms

James Ethridge, Gregory Ditzler and Robi Polikar

***Abstract*—Using a machine learning algorithm for a given application often requires tuning design parameters of the classifier to obtain optimal classification performance without overfitting. In this contribution, we present an evolutionary algorithm based approach for multi-objective optimization of the sensitivity and specificity of a ν-SVM. The ν-SVM is often preferred over the standard C-SVM due to smaller dynamic range of the ν parameter compared to the unlimited dynamic range of the C parameter. Instead of looking for a single optimization result, we look for a set of optimal solutions that lie along the Pareto optimality front. The traditional advantage of using the Pareto optimality is of course the flexibility to choose any of the solutions that lies on the Pareto optimality front. However, we show that simply maximizing sensitivity and specificity over the Pareto front leads to parameters that appear to be mathematically optimal yet still cause overfitting. We propose a multiple objective optimization approach with three objective functions to find additional parameter values that do not cause overfitting.**

***Index Terms*—multi-objective optimization, ν-SVM, evolutionary algorithms**

I. Introduction

PARAMETER optimization is often a time consuming task and generally requires a fair amount of prior knowledge about the data, and/or large repetitions of cross validation to appropriately select the parameters for a given classification algorithm (such as the number of hidden layer nodes of an MLP, the C parameter in C-SVM, the ν parameter in ν-SVM or the number of nearest neighbors in k-NN). The goal is to be able to optimally choose a set of parameters and be able to verify that the parameters will not cause the classifier to overfit to the training data. A classifier such as the k-NN only has one parameter, the number of nearest neighbors, to adjust/optimize, whereas the MLP has the number of hidden layers, number of hidden nodes, error goal and any parameters used in the backpropagation algorithms (perhaps the momentum term), to name a few. The selection of these parameters has a profound effect on the classification accuracy of the classifier, therefore it is critical for the designer to select these parameters through proper optimization techniques rather than applying ad hoc heuristic methods. To do so, two questions need to be answered: what objective function(s) should be optimized; and which optimization technique should be applied? Generally we must find at least two suitable objective functions to optimize [1]. Suitable objective functions could be complexity vs. accuracy or sensitivity vs. specificity [2]. A grid search can be utilized to optimize the parameters, however this approach is extremely time consuming and may not even always give optimal results when the grid is not densely sampled [3]. Evolutionary Algorithms (EA) have recently become popular and have been widely used in parameter optimization for several classifiers, such as the C-SVM. Not only are EA used for parameter optimization but feature selection as well for very high dimensional datasets [4]. It is important to note that, as shown by several previous efforts, the optimization of a single objective function, such as the validation error, may not yield an optimal classifier model [5]. As an alternative, the application of Multi-Objective Optimization (MOO) techniques using Evolutionary Algorithms (EA) has been explored in several works [1,6]. In the multi-objective case, the sensitivity and specificity of the classifier are frequently chosen as the objective functions for optimization. MOO-EAs have already been proposed in [1,2,5,7] for the selection of optimal SVM parameters. For example, the approach in maximizes the sensitivity and specificity for the C-SVM for a text classification problem [5]. These works have demonstrated that MOO-EA techniques are options for optimizing C-SVM parameters, however, very little has been done to properly check whether the classifier with these "optimal" parameter values is actually overfitting the training data.

The primary contribution of this work is the application of an EA, the Non-dominated Sorting Genetic Algorithm II (NSGA-II), for multi-objective optimization of the ν-SVM. This is presented as an alternative to previous research in which a set of optimal regularization and kernel parameters for the C-SVM algorithm are determined using an EA. The primary goal is to limit the bounds on the optimization parameter space in order to eliminate the heuristic selection of the upper and lower bounds on said parameters. We present the optimization of the ν-SVM using the standard Gaussian kernel function with a single spread parameter. A brief analysis of overfitting is also presented with the optimized parameters determined by the MOO-EA. The paper is organized as follows: Section II provides a background to the problem and previous work, Section III highlights the basic theory involved with the classifier and EA, Section IV highlights the results obtained with several

Manuscript received January 31, 2010 and revised on May 2, 2010. This work was supported by the National Science Foundation under Grant No: ECCS-0926159.

The authors are with the Dept. of Electrical and Computer Engineering at Rowan University and are a part of the Signal Processing & Pattern Recognition Laboratory, Glassboro, NJ, 08028, USA (e-mail: {ethrid60, ditzle53}@students.rowan.edu; polikar@rowan.edu).

publicly available databases, and Section V provides our discussion of the results and some concluding remarks.

II. Background

A considerable amount of prior work by many researchers has focused on the optimization of the parameters for *C*-SVM in the context of a MOO problem, applying EA to optimize the sensitivity and specificity of the *C*-SVM with respect to the *C* parameter [5]. The primary drawback of using the *C*-SVM in this context is that the *C* parameter has an infinitely wide dynamic range, bounded between $[0, \infty)$. Therefore, in most optimization approaches, one must select a somewhat arbitrary upper bound for *C*. For instance, in [5], the regularization parameter was selected with a range of 0-500 and then 0-5000 before the optimization algorithm could be applied. However, if enough information is not available to properly gauge the range of the regularization parameter, an arbitrarily selected range for the search can result in suboptimal solutions. In the absence of any prior knowledge, one might consider making the search range very large, but this will inherently increase the size of the parameter space which must be searched by the EA. In order to avoid the large range of the *C* parameter, we first propose replacing the *C*-SVM with the ν-SVM model originally presented by Schölkopf *et al*[8], as the ν parameter of the algorithms has an inherently finite range of [0 1]. Schölkopf *et al* [9] also showed that the parameter ν is a decreasing function of *C*. Therefore, a large *C* corresponds to a small ν (small margin) and a small *C* corresponds to a large ν (large margin). Thus, the ν parameter space eliminates the necessity to heuristically select an upper bound on the regularization parameter.

However, finding the optimal value that maximizes the classifier performance, without showing any care to avoid overfitting, also leads to a suboptimal solution. While the main objective function(s) may be maximized, there is little to no precaution taken in much of the previous work to prevent or test for overfitting.

III. Theory

A. ν-SVM

The ν-SVM is an alternate formulation of the traditional SVM, sometimes denoted as C-SVM. The ν-SVM and *C*-SVM are both maximum margin classifiers and use quadratic programming to solve for the support vectors, but regularization parameter (*C*), is replaced by ν, to control the margin in ν-SVM. The other difference between the two formulations is that the ν term allows for control of the fraction of support vectors and has a relationship to the fraction of training errors. It has been shown that this alternate formulation provides a more intuitive representation than the regularization parameter in *C*-SVM [10].

Eq. 1 shows the primal formulation of the *C*-SVM objective function, optimized subject to the constraints in Eq. 2 & 3, where $\mathbf{w}$ and b control the optimal hyperplane that maximizes the margin between the convex hulls of the two classes, *C* is the regularization parameter, $\mathbf{x}_n$ is the *n*th data instance, ξ_n is the corresponding slack parameter, t_n is the true label (-1 or 1), $\phi(\mathbf{x}_n)$ is the kernel function (if used) applied to the instance $\mathbf{x}_n$, and *N* is the total number of samples in the training data. This primal formulation of the *C*-SVM can be converted to its dual, and optimized using quadratic programming. The result of this optimization is a set of positive Lagrange multipliers that can be used to solve for $\mathbf{w}$ and b. The implementation of the C-SVM is done using the linear equation, $y(\mathbf{x}_m) = \mathbf{w}^t\phi(\mathbf{x}_m) + b$. The primary free parameters here are the choice of *C* (and the kernel option, if the SVM is to be used to implement a nonlinear decision boundary). Recall that *C* is the tradeoff between the slack variables penalty and the size of the margin. It also happens to be the upper bound on the Lagrange multipliers (α) in the dual formulation. If *C* is large, then a large penalty is assigned to the instances that cross or fall into the margin of the SVM and vice versa.

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \tag{1}$$

$$t_n(\mathbf{w}^t\phi(\mathbf{x}_n) + b) - 1 + \xi_n \geq 0 \tag{2}$$

$$\xi_n \geq 0 \tag{3}$$

In ν-SVM, the regularization parameter is replaced with $\nu \in [0,1]$, where ν is an upper bound on the fraction of margin errors. The margin errors come from all instances with $\xi_i > 0$ which are either training errors or examples that lie within the margin on the correct side of the decision boundary. The primal problem for the ν-SVM is shown in Eq. 4 subject to constraints in Eq. 5, 6 & 7. Note that *C* has been replaced with ν and ρ in this formulation, where $\rho \in [0,1]$ controls the width of the margin, which is equal to $2\rho/\|\mathbf{w}\|$.

$$\min_{\mathbf{w},\boldsymbol{\xi},\rho,b} \tau(\mathbf{w},\boldsymbol{\xi},\rho) = \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho + \frac{1}{N}\sum_{n=1}^{N}\xi_n \tag{4}$$

$$t_n(\mathbf{w}^t\phi(\mathbf{x}_n) + b) - \rho + \xi_n \geq 0 \tag{5}$$

$$\xi_n \geq 0 \tag{6}$$

$$0 \leq \rho \leq 1 \tag{7}$$

The minimization problem in Eq. 4 can be converted to its dual and can be solved via quadratic programming by maximizing Eq. 8, subject to constraints in Eq. 9, 10 and 11, where α_n are the Lagrange multipliers, $t_n \in \{\pm 1\}$ are the class labels and $k(\mathbf{x}_m, \mathbf{x}_n)$ is the kernel function (if used for nonlinear problems).

$$\tilde{L}(\alpha) = -\frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N}\alpha_m\alpha_n t_m t_n k(\mathbf{x}_m, \mathbf{x}_n) \tag{8}$$

$$0 \le \alpha_n \le \frac{1}{N} \tag{9}$$

$$\sum_{n=1}^{N} \alpha_n t_n = 0 \tag{10}$$

$$\sum_{n=1}^{N} \alpha_n \ge \nu \tag{11}$$

It can be shown that the lower bound of the fraction of the support vectors also happens to be ν. The clear advantage to using the ν-SVM formulation is that this free parameter is now bounded $\nu \in [0,1]$ rather than $C \in [0, \infty)$. Furthermore, by considering the optimal set of Lagrange multipliers, $\boldsymbol{\alpha}$, as a function of parameters, the relationship between ν and C can be shown to be [9]:

$$\lim_{C \to \infty} \frac{\sum_{n=1}^{N} \alpha_n}{CN} = \nu_{min} \ge 0 \quad and \quad \lim_{C \to 0} \frac{\sum_{n=1}^{N} \alpha_n}{CN} = \nu_{max} \le 1 \tag{12}$$

which implies that on any given problem with any given kernel, there exists an interval $[\nu_{min}, \nu_{max}]$, where $0 \le \nu_{min} \le \nu_{max} \le 1$, which maps to $C \in [0, \infty)$.

Thus, the ν parameter space has a tighter dynamic range and therefore a narrower search space than that of the *C*-SVM. Also, it has been shown in [9] that some interval of values in the *C* parameter space often map to a single value of ν. Thus, the ν-SVM eliminates the possibility of redundant parameters in the initial population of the EA, which could negatively impact the performance of the optimization algorithm.

Clearly, for nonlinearly separable problems, we also need to choose a kernel function and its associated free parameters. For the purposes of this paper, we focus on the commonly used radial basis function.

$$\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle = k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\sigma^2} \right) \tag{13}$$

The free parameter of this kernel is the choice of σ, which controls the spread of the radial basis function. Clearly, varying the spread can have a drastic effect on the function's reach: if the σ is very small then the kernel becomes very focused at a particular point. On the other hand, if σ is very large, the kernel becomes too broad. Therefore, we would like to use an EA to optimize σ and ν by generating a Pareto optimality set that will give the user a series of feasible solutions that can be used for their classification problem.

B. Multi-Objective Optimization

The goal in multi-objective optimization (MOO) is to simultaneously optimize two or more (possibly) conflicting objective functions subject to a set of predefined constraints. For a set of m objective functions, to be optimized with respect to a set n variables $\mathbf{x} = [x_1, x_2, \dots, x_n]$, the formulation is given as follows:

$$\min_{\mathbf{x}} (f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), \dots, f_m(\mathbf{x})) \tag{14}$$

$$subject\ to$$

$$g_i(\mathbf{x}) \le 0 \tag{15}$$

$$h_j(\mathbf{x}) = 0 \tag{16}$$

$$\mathbf{x}_{lower} \le \mathbf{x} \le \mathbf{x}_{upper} \tag{17}$$

where $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ are the *i*th and *j*th inequality and equality constraints, respectively. Each parameter may also be constrained between some lower and upper bound.

A set of optimality conditions for the multi-objective case, known as Pareto optimality conditions, define a set of feasible, non-dominated solutions in the objective space. A solution, $f(\mathbf{x})$, is said to be non-dominated if for any other feasible objective vector, $f(\mathbf{x}^*)$:

$$f_i(\mathbf{x}) \le f_i(\mathbf{x}^*);\ \forall i = 1, 2, 3 \dots m \tag{18}$$

and there exists at least one objective such that:

$$f_j(\mathbf{x}) < f_j(\mathbf{x}^*);\ j \in 1, 2, 3 \dots m \tag{19}$$

This is formally written as $f(\mathbf{x}) \prec f(\mathbf{x}^*)$. The Pareto front is then defined as the set of all such non-dominated points.

C. Application for NSGA-II Parameter Selection

In cases where the dimensionality of the MOO problem is high, or where the solution is non-convex, an analytical solution that produces the entire Pareto front may become intractable. In such cases, it is desirable to approach the MOO problem in a manner that generates a representative subset of the Pareto front. The application of evolutionary algorithms has become a popular approach to the aforementioned problem due to their ability to search for multiple Pareto optimal solutions concurrently as a result of the inherent parallelism in the algorithm [5].

The NSGA-II algorithm is an elitist EA that was developed to solve MOO problems with an emphasis on reduced computational complexity and diversity preservation [11]. The use of elitism in EAs applied to MOO problems has been shown to improve performance by allowing non-dominated solutions to persist across generations [12]. Diversity preservation ensures that the EA maintains a good spread of solutions along the generated Pareto front. Diversity preservation is implemented in NSGA-II using the crowded comparison operation; a crowding metric which is free of user defined parameters present in other algorithms [11]. Finally, the worst case computational complexity of the NSGA-II algorithm is

$O(MN^2)$, where M is the number of objectives and N is the population size. Thus the algorithm boasts a lower complexity than other MOO-EAs, such as Strength Pareto Evolutionary Approch-2 (SPEA-2) [13].

When applying the NSGA-II algorithm to the optimization of ν-SVM parameters, we need to determine the parameter space and the fitness function to be optimized. The parameter space defined for the ν-SVM is given by the ν parameter and the parameters associated with the chosen kernel, the spread parameter, σ, for the Gaussian kernel. The fitness function to be optimized by the NSGA-II algorithm is then the sensitivity and specificity for the generated classifier.

The formulation for this MOO problem is given by:

$$\min_{\nu,\sigma}(-f_1(\nu,\sigma), -f_2(\nu,\sigma)) \tag{20}$$

$$s.t.$$

$$0 \leq \nu \leq 1 \tag{21}$$

$$0 < \sigma < \infty \tag{22}$$

where, $f_1(\nu,\sigma)$ is the sensitivity and $f_2(\nu,\sigma)$ is the specificity of the classifier on a given data set.

IV. Results

The results presented in this section use 5-fold cross-validation on each dataset to independently evaluate the generalization of the sensitivity and specificity on each database. The Pareto optimality fronts are presented along with an interpretation of the curves. All databases were preprocessed to normalize all features to $[-1,1]$ range, which allows us to limit the search region of the spread parameter. A random label test was also performed to check whether the classifier is overfitting to the data with the parameters determined by the EA. This technique works well when the data is uniformly distributed in feature space.

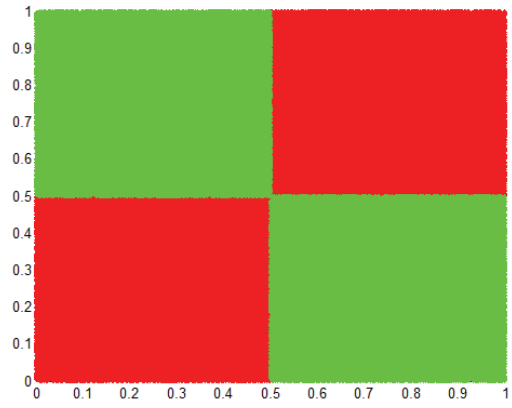

Fig. 1. Checkerboard dataset feature space.

Four datasets were used, whose names and cardinalities can be found in Table I. All databases, with the exception of the checkerboard dataset (which can be easily generated), are publically available from the UCI Machine Learning Repository [14]. The breast cancer database contains 10 features that come from a digitized image of a fine needle aspirate (FNA) of a breast mass which is used to classify malignant and benign diagnosis. The ionosphere database contains 34 continuous features and diagnoses of a feature that shows "good" or "poor" evidence of structure in the ionosphere. The Parkinson's dataset contains biomedical voice measurements with 23 features used to diagnose Parkinson's patients. The checkerboard dataset is a synthetic binary classification problem with zero rotation and four squares (see Figure 1). This problem reduces down to the classical non-linear XOR separation problem. The checkerboard training data was varied between 100 and 200 random samples and the testing dataset is 5000 random samples. All datasets described above are relatively balanced datasets, but it should be noted that not all of the databases have an equal number of examples from each class.

TABLE I. DATASETS USED

Dataset	*# of instances*
ionosphere	351
checkerboard	100/200
breast cancer	569
Parkinson's	197

A. Optimality Search Results

The resulting Pareto optimality curves for the Parkinson's, ionosphere, breast cancer, and checkerboard database can been seen in Figure 2, 3, 4, and 5, respectively. A variation of the checkerboard dataset is also shown in Figure 6. This second checkerboard test uses 200 examples in the training set rather than 100. The bold points in the figures – corresponding to the optimal values that actually cause overfitting – will be discussed later in the next section in more detail.

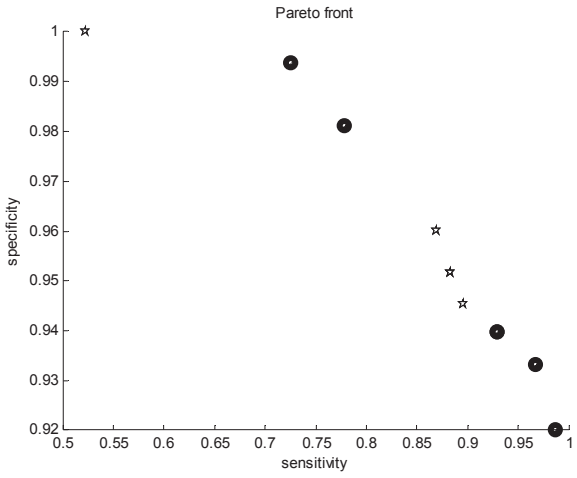

Fig. 2. Parkinson's Pareto optimal set.

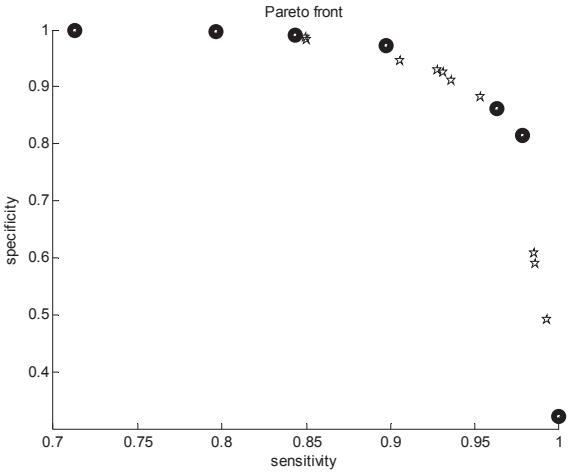

Fig. 3. Pareto optimal set for the Ionosphere dataset.

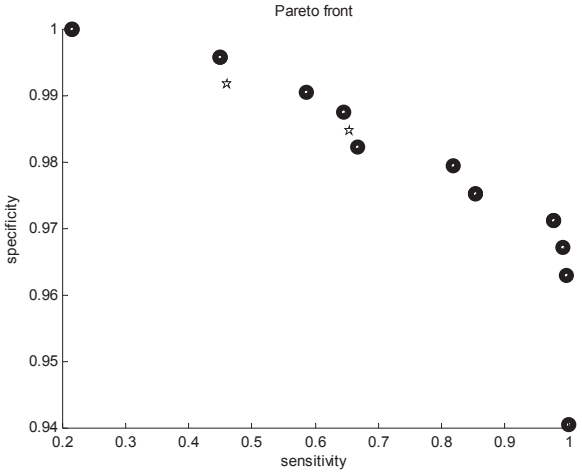

Fig. 4. Pareto optimal set for the breast cancer dataset.

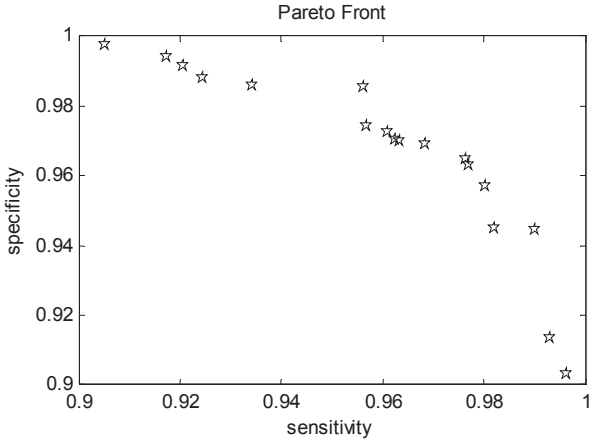

Fig. 5. Checkerboard (m=100) Pareto optimal set.

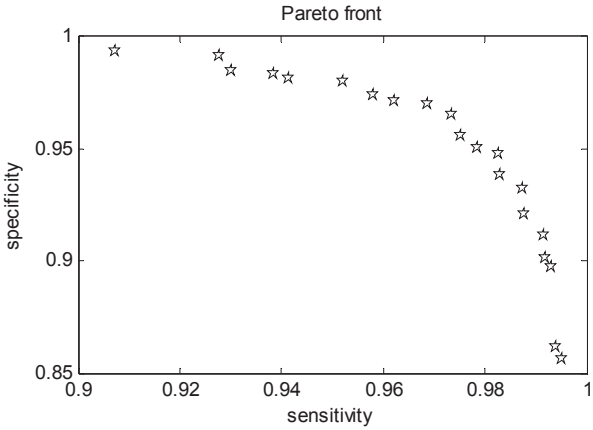

Fig. 6. Checkerboard (m=200) Pareto optimal set.

The maximization of the sensitivity and specificity returns a set of optimal values for ν and σ known as the Pareto optimality set. Each of these parameters was optimized such that no other point could maximize one of the objective functions without decreasing another. Each set provides a range of parameters that can be used to obtain a good balance between the sensitivity and specificity metrics on each dataset. However, there may be points along each curve that are better choices than others (i.e. some parameters may cause overfitting), as we discuss later in this section.

Table II, Table III, Table IV and Table V display some of the optimal parameters chosen from the Pareto set for the Parkinson's, ionosphere, breast cancer, and checkerboard datasets, respectively. Several interesting observations can be made from these tables. For example, there is very little variation in the selection of ν for the ionosphere database, which appears to indicate that this is the only possible solution, however this was not observed in other databases. The breast cancer and checkerboard datasets provided a wider spread of SVM parameters (both ν and σ) that can be used to maximize sensitivity and specificity, as shown in Table IV and V. Recall, that a large value of ν corresponds to a small value *C*. This means that the margin is wide and a small penalty is applied to the slack variables in the original SVM formulation.

The results also show that the evolutionary algorithms are able to find sets of parameters that can maximize the sensitivity and specificity of the ν-SVM classifier on a dataset. However, these algorithm parameters (ν/σ) do not tell us whether the classifiers trained with these "optimal" values will cause overfitting. Once we have a set of parameters, we need to determine which set(s) of parameters are best for the given dataset. Clearly, we would like a set of parameters that provides a good balance between sensitivity and specificity, but it is also important that the classifier is not overfitting. Therefore, generalization accuracy on previously unseen data is also an important quality sought from the classifier. We would like to determine a method or set of methods that can be used to analyze which set of parameters are overfitting and to what extent.

TABLE II. ν-SVM PARAMETERS CHOSEN FROM THE PARETO OPTIMAL SET FOR THE PARKINSON' DATASET

ν	σ	sensitivity	specificity
0.2407	0.0061	0.7787	0.9813
0.2150	0.0081	0.9667	0.9333
0.1401	0.3689	0.8825	0.9517
0.1072	0.5229	0.9867	0.9201

TABLE III. ν-SVM PARAMETERS CHOSEN FROM THE PARETO OPTIMAL SET FOR THE IONOSPHERE DATABASE

ν	σ	sensitivity	specificity
0.6270	0.1214	0.9929	0.4915
0.6676	0.2301	0.9534	0.8827
0.6363	0.4643	0.9359	0.9103
0.6811	0.3387	0.8504	0.9826

TABLE IV. ν-SVM PARAMETERS CHOSEN FROM THE PARETO OPTIMAL SET FOR THE BREAST CANCER DATABASE

ν	σ	sensitivity	specificity
0.6988	0.0038	0.6677	0.9825
0.3333	0.0073	0.8183	0.9795
0.0859	0.1843	0.9978	0.9631
0.0521	0.0832	0.9778	0.9713

TABLE V. ν-SVM PARAMETERS CHOSEN FROM THE PARETO OPTIMAL SET FOR THE CHECKERBOARD DATASET

ν	σ	sensitivity	specificity
0.1595	0.3909	0.9569	0.9742
0.0972	0.5432	0.9820	0.9451
0.0604	0.3006	0.9763	0.9648
0.0444	0.3684	0.9610	0.9725

B. *Overfitting Analysis*

The commonly used *k*-fold cross-validation procedure allows us a better estimate of the generalization error, but does not conclusively determine if a classifier is overfitting. In order to determine whether optimal parameters may cause overfitting, we devised a random label test. In the random label test we assign labels to the data completely at random, and then train a classifier using the (optimal) parameters determined by the EA, on this data whose labels are randomly assigned. The classifier is then evaluated with the test data, whose labels are also randomly assigned. Under this experiment, we would expect the classifier to perform close to random guess. If the classifier performs particularly well on the randomly labeled data, then this outcome shows that the classifier is clearly overfitting. We decided to let $\pm 15\%$ $(45 - 65\%$ $)$ of a random guess to indicate if a classifier was overfitting with an optimal set of parameters from the MOO task. The large range was selected because we are interested in determining what sets of parameters were causing the classifier to blatantly overfit to the data.

TABLE VI. RANDOM LABEL TESTING PERFORMED ON THE CHECKERBOARD DATASET

ν	σ	error
0.1411	0.5920	0.4900
0.0537	0.3450	0.5200
0.0537	0.3451	0.4650
0.0566	0.3623	0.4850
0.0550	0.3776	0.5300
0.0568	0.3717	0.5300
0.0736	0.3971	0.4750

Table VI and VII provides the results of the random label test for the checkerboard and Parkinson's dataset, respectively. In Table VI, the error after applying random labels to the data and training a classifier using different values of ν and σ are very close to random chance for this two class checkerboard problem. This is an indication that the classifiers trained with the checkerboard dataset – using the parameters selected from the Pareto optimality front - are *not* overfitting. In fact all of the random label assignment tests for the checkerboard dataset displayed results that were approximately within $\pm 5\%$ of the random guess. On the other hand, as we discuss next, checkerboard was the only database presented in this paper for which the optimally determined parameters did not cause the classifier overfit. We believe this is due to the relative simplicity of the problem and the data being uniformly sampled from feature space.

TABLE VII. RANDOM LABEL TESTING PERFORMED ON THE PARKINSON'S DATASET

ν	σ	error
0.1072	0.5229	0.5983
0.1254	0.4173	0.5726
0.1401	0.3689	0.5128
0.1914	0.0058	0.9972
0.2520	0.9603	0.5185
0.1072	0.5229	0.5983

The spread parameter of the RBF kernel provides another indicative measure of overfitting of the ν-SVM. A classifier that is overfitting the data will typically have a very small spread parameter, thus becoming very focused at a particular point. For example, the spread parameters for the first two entries in Table IV in the breast cancer database are very small. The fourth entry in Table VII also has very small spread parameter compared to the others and the overall error after applying the random label test is 99.7% (which, with the flip of labels, effectively represents a 99.7% accuracy). This is a clear indication that the parameters in the fourth entry of Table VII are causing overfitting. Other optimal sets shown in Table VII do not have nearly as small σ , and correspondingly, as extreme error rate. In fact most of them are just around random chance, as we would normally expect from the a properly trained classifier on a

random label test. Yet, with the exception of the checkerboard dataset, all other datasets resulted in optimal parameters that overfit the data. For instance, the first two examples in Table II and IV have very small spread parameters, whose random label errors were between 0.96-1.0, a clear indication of overfitting. The breast cancer database was by far the worst by having nearly all the optimal parameters achieve a very high classification rate with the randomly assigned labels.

The bold points in the Pareto front curves indicate examples that appear to yield a classifier that has a high/low classification rate on a randomly labeled dataset. The threshold was set such that the overfitting examples were above 65% after a random label test was applied. Therefore, we argue that each Pareto optimal point should be evaluated for overfitting either by performing this test on the classifier or by some other means of the designer's choice.

C. Modified Multi Objective Function

In this section, we present the preliminary results of adding an additional objective function to be minimized in the optimization problem. The first and second objective functions are still sensitivity and specificity, respectively. The third objective function is the absolute value of the difference between the random label error and a random guess for a two class problem. Specifically, $f_3(\nu,\sigma) = |0.5-\epsilon|$ where ϵ is the generalization error produced by the ν-SVM with the ν and σ parameters selected by the EA after cross validation.

$$\min_{\nu,\sigma}(-f_1(\nu,\sigma), -f_2(\nu,\sigma), f_3(\nu,\sigma)) \tag{23}$$

$$s.t.$$

$$0 \le \nu \le 1 \tag{24}$$

$$0 < \sigma < \infty \tag{25}$$

Table VIII and Figure 8 show the test results using ν-SVM parameters determined with the third objective function optimization, on the worst case breast cancer dataset. Unlike the previous case, where most optimal solutions caused overfitting, the new optimization provides several solutions that do not cause overfitting. The algorithm also makes it trivial to detect overfitting – when it happens – with a zero error on random label test. As expected, these cases also correspond to very small spread values. All other sets of parameters result in a random label test error close to 0.5, indicating that the classifier is not overfitting.

V. Conclusion

We have described an application of EA for the maximization of sensitivity and specificity of the ν–SVM, using a multi-objective optimization approach. The obvious benefit of using this approach is that optimization techniques can be applied to determine parameters rather than heuristics thus providing a sound justification for using the set of parameters for a specific dataset.

TABLE VIII. RANDOM LABEL TESTING PERFORMED ON THE BREAST CANCER DATASET

ν	σ	error
0.1106	0.5598	0.4938
0.1368	0.7809	0.4903
0.1438	0.1968	0.5131
0.1637	0.6205	0.4938
0.1676	0.6047	0.4815
0.1919	0.0069	0
0.3049	0.0023	0
0.5437	0.0015	0
0.0411	0.0528	0.4534

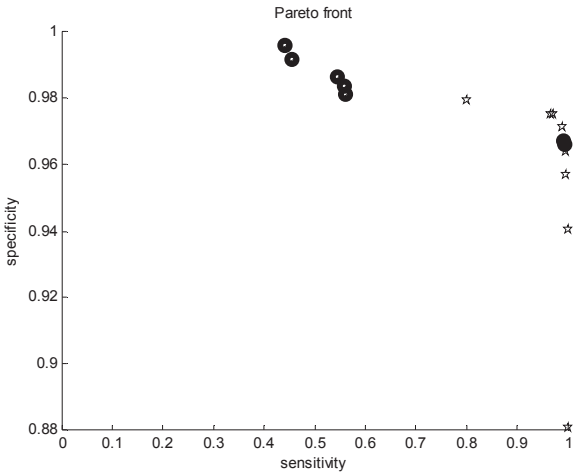

Fig. 7. Pareto optimal set for the breast cancer dataset with the modified optimization problem.

The ν–SVM was chosen in this work because the ν parameter in the optimization of the classifier is bound between [0,1] unlike the more commonly used C-SVM which does not have this bound. The initial results show that there is a diverse set of parameters that can be used to maximize sensitivity and specificity. However, it may be more advantageous for a designer to choose one set of parameter over another as some sets of parameters may cause the classifier to overfit the training data.

We have then described a simple procedure that can be applied to determine if the classifier is overfitting, by assigning random labels to the data. One would expect the error of a classifier trained/tested with randomly labeled data to be near random guess. The initial results have shown that several of the solutions in the Pareto optimal set do in fact cause overfitting. This was especially evident in the breast cancer dataset. Nearly all the solutions suggested by the Pareto set caused overfitting, whereas the checkerboard dataset had few indications of overfitting. This leads us to the conclusion that the optimal parameter sets should be checked for overfitting regardless of the optimization approach used, but certainly for the commonly used EA approaches. Adding a third objective function, as the absolute value of the difference between the random label error and a random guess for a two class problem appeared

to provide several parameters that did not cause overfitting. Our future efforts will focus in additional experiments to determine the theoretical and experimental behaviors of this approach.

REFERENCES

[1] C. Igel, "Multiobjective Model Selection for Support Vector Machines," in *Proc. of the 3rd Int. Conf. on Evolutionary Muilt Criterion Optimzation*, 2005, pp. 534-546.

[2] T. Suttorp and C. Igel, "Multi-Objective Optimization of Support Vector Machines," *Multi-Objective Machine Learning*, pp. 199-220, 2006.

[3] S. Lavalle and M. Branicky, "On the relationship between classical grid search and probabilistic roadmaps," *International Journal of Robotics research*, vol. 23, pp. 673-692, 2002.

[4] H. Frohlich, O. Chapelle, and B. Scholkopf, "Feature selection for support vector machines using genetic algorithms," *International Journal on Artifical Intelligence Tools*, vol. 13, no. 4, pp. 791-800, 2004.

[5] C. Chatelain, S. Adam, Y. Lecourtier, L. Heutte, and T. Paquet, "Multi-Objective Optimization for SVM Model Selection," in *9th International Conference on Document Analysis and Recognition (ICDAR)*, Curitiba, Parana, Brazil, 2007, pp. 427-431.

[6] Y. Jin and B. Sendhoff, "Pareto-Based Multi-Objective Machine Learning : An Overview and Case Studies," *IEEE Transactions on Systems, Man and Cybernetics: Part C*, vol. 38, no. 3, pp. 397-415, 2008.

[7] C. H. Wu, G. H. Tzeng, Y. J. Goo, and W. C. Fang, "A real-valued genetic algorithm to optimize the parameters of the support vector machine for predicting bankruptcy," *Expert Systems with Applications*, vol. 32, no. 2, pp. 397-408, 2007.

[8] B. Scholkopf, A. Smola, R. Williamson, and P. Bartlett, "New Support Vector Machine Algorithms," *Neural Computation*, vol. 12, pp. 1207-1245, 2000.

[9] C. C. B. Scholkopf and C. Lin, "Training [nu]-Support Vector Classifiers: Theory and Algorithms," *Neural Computation*, vol. 13, no. 4, pp. 2119-2147, Apr. 2001.

[10] R. Bergman, M. Griss, and C. Staelin, "A personal email assistant," HP Laboratories, Palo Alto, CA, Technical Report HPL-2002-236, 2002.

[11] D. Klayanmoy, S. Pratap, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, Apr. 2002.

[12] E. Zitzler, D. Kalyanmoy, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results", *Evolutionary Computation*, vol. 8, no. 2, pp. 173-195, Jun. 2000.

[13] A. Konak, D. Coit, and A. Sith, "Multi-optimization using genetic algorithms: A tutorial," *Reliability Engineering and System Safety*, vol. 91, pp. 992-1007, Jan. 2006.

[14] A. Asuncion. and D. J. Newman. (2007, Jan.) UCI Machine Learning Repository. [Online]. http://archive.ics.uci.edu/ml/

[15] J. Arora, *Introduction to Optimum Design*, 2nd ed. Elsevier Inc, 2004.

[16] C. L. B. S. P. Chen, "A tutorial on [nu]-support vector machines," *Appl. Stochastic Models Bus. Ind.*, vol. 21, no. 2, pp. 111-136, Apr. 2005.

[17] H. Zhao, "A multi-objective genetic programming approach to Pareto optimal decision trees," *Decision Support Systems*, vol. 43, no. 3, pp. 809-826, Dec. 2006.