EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
Vol. 9, No. 2, 2016, 216-230
ISSN 1307-5543 – www.ejpam.com

Regularized SVM Classification with a new Complexity-Driven Stochastic Optimizer

J. Andrew Howe[1,2,*], Hamparsum Bozdogan[3]

[1] *Risk Dynamics Consultancy, Istanbul, Turkey*
[2] *KAPSARC, Riyadh, Saudi Arabia (affiliated after article accepted)*
[3] *University of Tennessee, Department of Business Analytics & Statistics, Knoxville, Tennessee, USA*

Abstract. Given a multivariate dataset composed of data from different known sources or processes, how can we create a rule to separate the data, and classify any future data? Kernel discriminant analysis is one of many supervised learning techniques that handle this problem. Recently, in this and other knowledge discovery problems, kernel methods have gained popularity. This is somewhat ironic as another common theme is variable reduction, and kernel methods actually inflate dimensionality. Due to the substantial benefits of processing "kernelized" data, this is excusable - kernel methods frequently outperform traditional classification techniques for real data when the classes are not easily separable. In performing kernel discriminant analysis, there are two main issues that we address in this article. The first is that, in the literature, the question of which kernel function to use is often subjectively selected *a prior*, or determined by cross-validation with the sole objective of maximizing classification performance. Secondly, after obtaining discriminant functions or support vectors to classify a dataset, how do we know which of our variables are most responsible for, and important to, the classification? In this research, we develop a new regularized algorithm that simultaneously selects the kernel function and subset of original variables. Our algorithm, a hybrid of cross-validation and the genetic algorithm, does this by optimizing a function that rewards correct classification while penalizing model complexity and misclassification.
We report results on three real datasets, including data from a medical imaging study. For the latter, we obtained an impressively low misclassification rate of 0.3%, while reducing the number of features from $p = 20$ to $p^* = 6$.

2010 Mathematics Subject Classifications: 62H30, 68T10

Key Words and Phrases: Supervised classification, Discriminant analysis, Support vectors, Information criteria, Feature selection, Stochastic optimization, Reproducing kernel Hilbert space, Machine learning

*Corresponding author.

Email addresses: ahowe42@gmail.com (J. Howe), bozdogan@utk.edu (H. Bozdogan)

1. Introduction

Logistic regression is a well-known form of nonlinear regression in which responses can take on values of 0 or 1 (or any binary format). From a different perspective, we can see the binary responses as class labels for data from two groups. This leads us to the concept of discriminant analysis, which is closely related to logistic regression. In general, the goal of discriminant analysis is to determine data groupings that minimize the variability within the groups and maximize the variability between the groups. Differently put, given known class labels for the data, the goal is to minimize the probability of misclassification - it is a method of **supervised learning**. When we perform kernel discriminant analysis (KDA), this does not change. To use kernel discriminant analysis, one merely has to apply a kernel function to the data, then perform the usual analysis. This does, however, present two issues which we address in this research.

- There are many possible kernel functions that can be applied to perform the nonlinear map into higher-dimensional feature space. The choice of the kernel function can have a substantial impact on the analytical results. For example, use of a linear kernel on data that is inherently nonlinear will likely inflate the classification error. For univariate or bivariate data, it may be easy to determine nonlinearity, but what of ten variables? In most relevant research with kernel methods, the kernel function is either selected *a priori* or using cross-validation to minimize the testing classification error. How do we choose the "best" kernel function consistent with the principal of Occam's Razor?

- For any classification method, there is often value in knowing which variables contribute most to the separation of the classes. Of course, this is generically true about all statistical data mining / machine learning techniques. However, with kernel methods, after we apply the kernel function, it is impossible to perform any meaningful feature selection analysis. Perhaps we are making ten hopefully predictive, and costly, measurements on a process but we can get very low classification errors only using three of them? Why take unnecessary measurements for a model that is overly complex when we can have a model that is both accurate and parsimonious? How do we determine which variables are contributing the most to our discriminant functions in the kernel space?

Here we propose a hybrid optimization algorithm to simultaneously perform kernel selection and feature selection. Our algorithm uses a combination of cross-validation with the genetic algorithm. The optimization objective is a special form of Bozdogan's $ICOMP$ [12] developed for sparse classification methods such as KDA. Following this introduction, we provide background details of regularized KDA with support vector machines. Section 3 describes our hybrid algorithm, and numerical results on real datasets are shown in Section 4. We finish with some final thoughts in Section 5.

2. Regularized Kernel Discriminant Analysis with Support Vectors

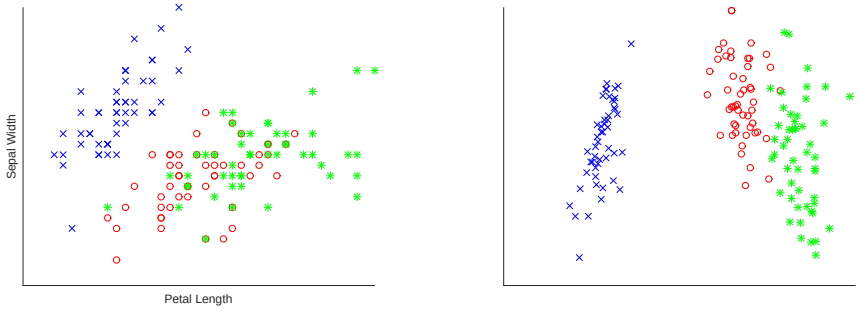
2.1. Kernels

Reproducing Kernel Hilbert Spaces were initially developed by the mathematician Aronszajn [2]. Assuming they meet Mercer's conditions, kernel functions correspond to a nonlinear map into a higher dimensional feature space $\mathbb{F}$ and then taking the dot product in this space. As an example, consider the map $\phi : \mathbb{R}^2 \to \mathbb{R}^3$, defined as $\phi\left(\left[x_1, x_2\right]'\right) = \left[x_2^2, \sqrt{2}x_1x_2, x_2^2\right]'$. For two vectors x_i and x_j, we have

$$\begin{aligned}\phi\left(x_i\right)'\phi\left(x_j\right) &= \left[x_{i2}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2\right]\left[x_{j2}^2, \sqrt{2}x_{j1}x_{j2}, x_{j2}^2\right]' \\ &= \left[x_{i2}^2x_{j2}^2, 2x_{i1}x_{i2}x_{j1}x_{j2}, x_{i2}^2x_{j2}^2\right] \\ &= \left(x_i'x_j\right)^2\end{aligned}$$

This last equality is called the quadratic kernel (with no intercept), which we may denote as $K\left(x_i, x_j\right)$; by computing the quadratic kernel function, we avoid performing the map. This is called the *kernel trick*. For a dataset x with n rows and p variables, the data is translated into a square matrix of size n in the feature space, in which every possible pair of points is evaluated. Note that this translation is neither one-to-one, nor onto - application of a kernel function is an non-invertible process. Once the data has been classified in the feature space, we have no way to go back to the original data space and judge the value of individual variables.

While kernel functions execute a non-intuitive process of dimensional inflation, the frequent superior performance of analysis with kernels is well documented in a variety of methods (KDA, KPCA, KLR, ...). At least part of this success is due to the way functions of distances between all pairs of observations end up as observations in the feature space. This has the effect of extracting much more information from the data. Related data tends to become more closely compacted, and unrelated data tends to become more separated. Visualization of this fact can be seen in Figure 1. In the right pane, note how the confounding between the *Versicolor* and *Virginica* observations is lessened, and how the *Setosa* group has pulled away from the others.



(a) Iris Data (b) 1st two Variables after Kernel Trick

Figure 1: Demonstrating Separation in Kernel Space; x is Setosa, o is Versicolor, and $*$ is Virginica.

The set of kernel functions which we use in this research is composed of the nine most common, as listed in Table 1. For each kernel, we indicate the parameters we used. It is generally

Table 1: Kernel Functions Available.

Function	Form	Parameters	Binary
Linear Polynomial	$\left(XX' + b\right)^a$	$a = 1, b = 0$	0001
Quadratic Polynomial	$\left(XX' + b\right)^a$	$a = 2, b = 1$	0010
Cubic Polynomial	$\left(XX' + b\right)^a$	$a = 3, b = 1$	0011
Exponential	$\exp\left(-\frac{1}{2a^2}\sqrt{\|\|X - X\|\|^2}\right)$	$a = 1$	0100
Gaussian	$\exp\left(-\left(\frac{1}{a^b}\|\|X - X\|\|^2\right)^c\right)$	$a = 2, b = c = 1$	0101
Homogenous Poly. (homo)	$\left(\frac{1}{a^2}XX'\right)^b$	$a = 1, b = 2$	0110
Laplace	$\exp\left(-\sqrt{\frac{1}{a^2}\|\|X - X\|\|^2}\right)$	$a = 1$	0111
Cauchy	$\left(\mathbf{1} + \frac{1}{a}\|\|X - X\|\|^2\right)^{-1}$	$a = 1$	1000
Inverse Multi-Quadric	$\left(\|\|X - X\|\|^2 + a^2\right)^{-\frac{1}{2}}$	$a = 1$	1001

difficult to know *a priori* which kernel is best. Yet, this is one decision that can have a large impact on the performance of KDA. Some researchers simply select a kernel function *a priori*. The more usual approach is to simply apply KDA with a variety of kernel functions and select whichever gives best results on the sample data. Cross-validation sampling is used to limit the risk of fitting to random noise in specific subsets. However, this approach only considers classification performance, and ignores model complexity. Optimal selection of the parameters is another important consideration in KDA. Liberati *et al.* [12] proposed a data-based technique to simultaneously optimize selection and parametrization of the kernel function. In this research, we have opted not to attempt to optimize the parameters.

2.2. Binary Discriminant Analysis

Our data x is composed of n realizations of p continuous measurements. Accompanying X is a vector of class assignments $y \in [1,2]$. Having $y_i = 1$ indicates that x_i is an element of the first group. After mapping the data into the feature space $\mathbb{F}$, the goal is to find a direction

$$\psi = \sum_{i=1}^{n} \alpha_i \phi\left(x_i\right) = \left\langle \alpha, x_i \right\rangle \tag{1}$$

with weight vector α that maximizes the Fisher criterion

$$J_F(\alpha) = \frac{\alpha' \hat{\Sigma}_B \alpha}{\alpha' \hat{\Sigma}_W \alpha}. \tag{2}$$

$\hat{\Sigma}_B$ indicates the *between-group covariance matrix*, and $\hat{\Sigma}_W$ is the *within-group covariance matrix*, shown in (3) and (4), respectively.

$$\hat{\Sigma}_B = \left(\overline{x_1} - \overline{x_2}\right)' \left(\overline{x_1} - \overline{x_2}\right) \tag{3}$$

$$\hat{\Sigma}_W = \frac{1}{n} W = \frac{1}{n} \sum_{k=1}^{2} I_{y_i = k} \left(x - \overline{x_k}\right)' \left(x - \overline{x_k}\right) \tag{4}$$

$\overline{x_k}$ is the mean of observations belonging to class k, and $I_{y=k}$ is an indicator function which takes on the value 1 when the specific datapoint is in group k (0 otherwise). The binary kernel discriminant function and classifier are:

$$f\left(x_i\right) = \left\langle \alpha, K\left(x_i, x\right) \right\rangle + b, \text{ and} \tag{5}$$

$$q\left(x_i\right) = \begin{cases} 1 & f\left(x_i\right) \geq 0 \\ 2 & f\left(x_i\right) < 0 \end{cases}. \tag{6}$$

where the vector α is obtained by solving (2), and b is the intercept of the separating hyperplane, which passes through the midpoint of the class centroids.

2.3. Binary Support Vectors

Generalizing this further, we can rewrite (5) as

$$f\left(x_i\right) = \left\langle \alpha^*, k_s\left(x_i\right) \right\rangle + b^*,$$

where $k_s\left(x_i\right) = \left[K\left(x_i, s_1\right), K\left(x_i, s_2\right), \ldots, K\left(x_i, s_m\right)\right]$ is the vector of the i^{th} datapoint evaluated at the m support vectors, which form a subset of the data. This is the *support vector machine (SVM)*. Thus, optimization of the weights and intercept becomes the quadratic programming problem shown here.

$$\left(\alpha^*, b^*\right) = \min_{\alpha, b} \left[\frac{1}{2} \|\alpha\|^2 + C \sum_{i=1}^{n} \xi_i^d \right], \text{ subject to}$$

$$\begin{aligned}&\left\langle \alpha, x_i \right\rangle + b \geq 1 - \xi_i, i \in I_1,\\ &\left\langle \alpha, x_i \right\rangle + b \leq -1 + \xi_i, i \in I_2,\\ &C > 0, \xi_i \geq 0, i \in I_1 \bigcup I_2\end{aligned}$$

When $d = 1$, we say the SVM is *L1 soft margin trained*, otherwise, it's *L2 soft margin trained*. C is a regularization constant, and I_1 and I_2 are slack variables used to relax the inequalities for non-separable data.

2.4. Multi-class SVM

For data composed of $K > 2$ classes indexed by k, we consider a set of discriminant functions

$$f_k\left(x_i\right) = \left\langle \alpha_k, k_s\left(x_i\right)\right\rangle + b_k.$$

There are several ways to decompose the multi-class SVM, including *One-Against All (OAA)* and *One-Against-One (OAO)* - see [11].

The OAA decomposition works by trading the single multi-class problem for K binary SVM problems, where the binary state vector y_k' is

$$y_k' = \begin{cases} 1 & \text{for } y = y_k \\ 2 & \text{for } y \neq y_k \end{cases}.$$

For example, if we had $K = 4$ classes **A**, **B**, **C**, and **D**, OAA would solve four binary problems:

A vs **BCD**, **B** vs **ACD**, **C** vs **ABD**, **D** vs **ABC**

The multi-class classification rule used is then

$$q\left(x_i\right) = \max_{k=1,2,\ldots,K} f_k\left(x_i\right). \tag{7}$$

OAO, on the other hand, solves the multi-class problem by solving $K' = K\left(K-1\right)/2$ binary SVM problems, in which all pairs of classes are considered. The majority voting strategy shown in (8) is used to select the final class assignments. The vector vote$\left(x_i\right)$ indicates the frequency with which, from all K' binary SVM results, the i$^{\text{th}}$ datapoint was classified into each group.

$$\begin{aligned}\text{vote}\left(x_i\right) &= \left[v_1\left(x_i\right), v_2\left(x_i\right), \ldots, v_{K'}\left(x_i\right)\right]\\ q\left(x_i\right) &= \max_{y_i'=1,2,\ldots,K'} vote\left(x_i\right)\end{aligned} \tag{8}$$

Using the same groups **A**, **B**, **C**, and **D**, OAO solves the six binary SVMs

A vs **B**, **A** vs **C**, **A** vs **D**, **B** vs **C**, **B** vs **D**, **C** vs **D**

2.5. Hybridized Covariance Estimation

In broad terms, KDA means applying the appropriate calculations in Sections 2.2 through 2.4 after application of the kernel trick. When performing KDA, it can be usually expected that the within-group covariance matrix (4) won't be nonsingular or positive definite. Singularity of this kernel covariance matrix is a problem that has attracted many researchers, and many methods have been proposed to make the matrix well-conditioned. In [12], Liberati *et al.* introduced a hybridized covariance estimator $\hat{\Sigma}_{STA\_CSE}$ that joins the stabilization technique of Thomaz [15] with the *convex sum covariance estimator* shrinkage technique of press [14] and Chen [6]. There is an optional third step, which we apply to help regularize especially sparse and / or singular matrices. After computing the hybrid stabilized kernel matrix, we compute its singular values. $\hat{\Sigma}_{STA\_CSE}$ is then replaced with a diagonal matrix of some subset of the largest singular values as a *reduced rank approximation* $\hat{\Sigma}^{*}_{STA\_CSE}$. For the real datasets analyzed in this research, we kept the top 25 when further regularization was necessary.

3. Cross-Validation Genetic Algorithm for Optimization

The *Genetic Algorithm (GA)* is a stochastic search algorithm that borrows concepts from biological evolution [7–10]; it has been used successfully in a wide range of statistical modeling applications. The solution space for a problem is explored via an ensemble of strings which represent possible solutions. In the parlance of the GA, these solution strings are called *chromosomes*. Chromosomes are created by encoding solutions using a fixed, finite-length alphabet of symbols. Solutions for the GA are most typically coded as binary strings. Individual chromosomes are allowed to compete with each other *a la* natural selection, to create better solutions, with the goal to optimize some objective function. Operational details of the GA can be found in the above-mentioned sources, or any number of others.

For the problem of selecting a subset of p variables, a chromosome is a p-length vector such that each element represents the presence (1) or absence (0) of a specific variable. An example chromosome may be [10011001]; in this case, predictors 1,4,5,8 are selected while 2,3,6,7 are not. Our algorithm uses the GA to simultaneously determine the best subset of variables AND the best kernel function. We do this by coding the different kernel functions into a binary string, and appending this onto the feature selection string.

We have 9 kernels to choose from; 9 in binary representation requires 4 bits. Hence, the kernels are represented by binary strings from [0000] (linear polynomial) to [1001] (inverse multi-quadric). The allowed kernel binary codes are shown along with the kernel functions in Table 1. An obvious issue is that four binary digits can be used to represent all the counting numbers up to fifteen, and yet we only have nine kernels. Hence, the GA's crossover and mutation operators can both generate illegal strings representing kernels that we don't have.

When this occurs, we apply modular arithmetic to scale back the offending binary string. For example, say the kernel portion of a chromosome is [1100], encoding the nonexistent twelfth kernel. Applying modular arithmetic, we have $mod(12, 9) = 3 = [0011]$. Thus, we replace a string that is too large with its remainder when divided by [1001]. This repair mechanism will impart a slight bias to the six simplest kernel functions, but does not seem to

be have any systemic effect.

Hence, a chromosome in our algorithm is a $p+4$-length binary string representing a specific subset of features and kernel function. The genetic algorithm is allowed to progress for twenty generations, operating on an ensemble of twenty of these solutions, with the goal to minimize the objective function $ICOMP_{PERF}$.

3.1. Information Complexity Criteria for Supervised Classification

A logical next step from information theoretic model selection criteria such as Akaike's AIC and Schwartz's SBC, Bozdogan introduced and developed his information complexity criteria $ICOMP$ in [4, 5, (and others)]. $ICOMP$ typically penalizes models based on the maximal entropic complexity C_1 of the model covariance matrix [3], as opposed to functions of the number parameters estimated. This penalty allows $ICOMP$, shown in (9), to simultaneously penalize a model based on **lack-of-fit**, **lack-of-parsimony** and **profusion-of-complexity**.

$$ICOMP(\hat{\mathscr{F}}^{-1}) = \underbrace{-2\log L(\hat{\theta} \mid X)}_{\text{lack-of-fit}} + \underbrace{2C_1(\hat{\mathscr{F}}^{-1})}_{complexity} \tag{9}$$

Several versions of $ICOMP$ have been developed for various model selection problems. For the problem of kernel-based supervised learning, Liberati *et al.* [12] introduced $ICOMP_{PERF}$. This form of $ICOMP$ is conceptually based on the regression-basis of discriminant analysis:

$$ICOMP_{PERF} = \underbrace{n\log 2\pi + n\log\hat{\sigma}^2 + n}_{lack-of-fit} + \underbrace{2C_{1F}\left(\hat{\Sigma}^{*}_{STA\_CSE}\right)}_{complexity}, \tag{10}$$

where $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$. The y_i are the actual known group labels, and the $\hat{y}_i$ are the group labels predicted with KDA. The kernel methods used in KDA lead to orthogonal and highly sparse matrices, which can cause problems for the C_1 measure of complexity. Accordingly, $ICOMP_{PERF}$ uses a modified measure of complexity based on the *Frobenius norm* characterization of the entropic complexity C_F, shown in (11) as a function of the eigenvalues of the matrix.

$$C_{1F}(\cdot) = \frac{1}{4\overline{\lambda}^2}\sum_{i=1}^{p}\left(\lambda_i - \overline{\lambda}\right)^2 \tag{11}$$

By minimizing $ICOMP_{PERF}$, our algorithm can simultaneously **minimize classification error**, **maximize fit**, and **minimize model complexity**.

3.2. Cross-Validation within the GA: CVGA

Cross-validation (CV) partitioning of datasets is commonly used in machine learning to assess the predictive power of models. The model is estimated based on a *training set* of the data, then the model is used to classify a *testing set*. Classification rates are reported from the testing set. Observations are randomly grouped into training and testing sets based on a specified percentage (in this research, we use 20% for training, with the remaining 80% for testing).

To avoid bias due to the randomized partitioning, we could run the GA many times (called *replications*), each time with a new CV sample, but it would be very time consuming to do this on top of the GA. At the end, we'd need to combine the results from all the replications to determine the optimal model. Instead, we propose a hybrid algorithm that combines the GA with cross-validation, called *CVGA*. CVGA adds an extra layer of randomness to the GA in that every time the fitness of a chromosome is evaluated, it is based on a new cross-validation sample. Instead of nesting the GA within CV, CV is nested within the GA. This allows us to score solutions during optimization with no concern that our partitioning into training / testing for the GA was somehow not representative. However, the extra randomness within the CVGA does present a challenge.

Suppose in one iteration of the CVGA, we find the best solution to use variables 1 and 2 and the Gaussian kernel, with a score of -67. In a later generation, the same solution could result in a score of -23, and not be the best for that iteration. This variability is obviously due to the fact that the fitness of each chromosome is evaluated on different data. We have implemented a two-stage process to smooth out this variation.

First, in summarizing the results from the GA, we store the median $ICOMP_{PERF}$ and testing / training classification error rates for every unique solution (since good solutions will tend to be occur multiple times). The median is used instead of the average to ensure that the optimal solution selected at the end of the process is robust against cross-validations with outlying performances. Secondly, we actually run the GA 100 times, for a maximum of $100 \times 20 \times 20 = 40,000$ unique solutions evaluated. The median $ICOMP_{PERF}$ for each solution evaluated is then averaged over all 100 replications, and we compute the 95% empirical confidence intervals for the misclassification rates.

In summary, computing the fitness for each chromosome follows these steps:

(i) randomly partition the data into training and testing sets, keeping only the features selected by the chromosome

(ii) perform KDA on the training set, measuring the classification error, and the complexity of the reduced rank hybrid stabilized kernel matrix; this is for the complexity term of $ICOMP_{PERF}$ in (10)

(iii) use the KDA model to classify the data in the testing set, measuring the classification error, and the negative maximized likelihood; this is for the lack-of-fit term in (10)

(iv) join the two parts of $ICOMP_{PERF}$ together

4. Numerical Results

4.1. X-ray Burst Event Data

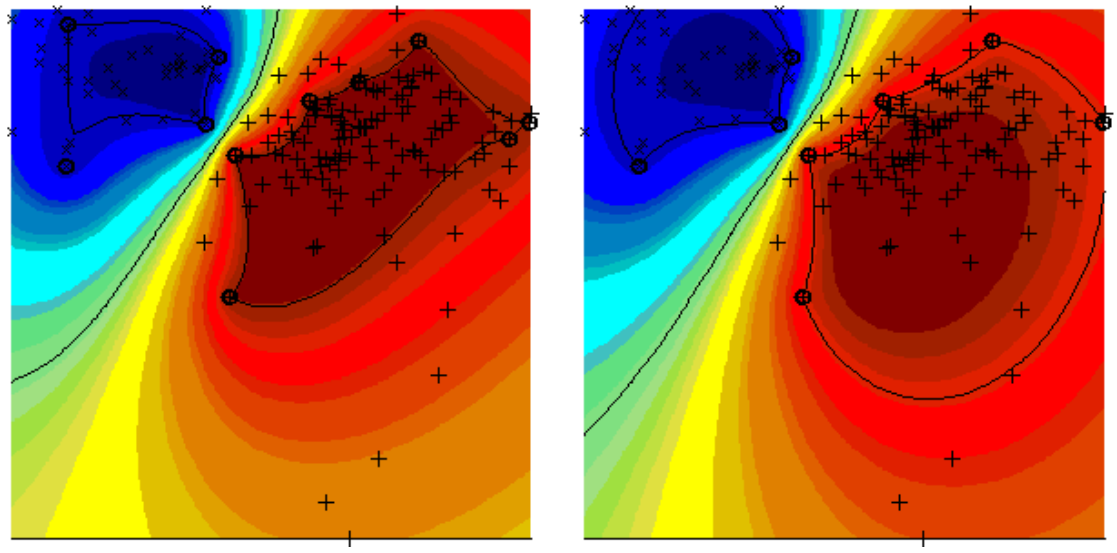
Our first example is an easy 2-dimensional dataset regarding X-ray bursts as gathered by the PHEBUS instrument on the Granat Satellite. This data was gathered with the purpose of predicting the effects of x-ray bursts on weather. There are two classes of data: the first class

with $n_1 = 132$ observations, and the second with the remaining $n_2 = 42$. Table 2 displays the four unique solutions chosen across by the CVGA.

Table 2: Models Selected for the X-ray Burst Data.

Subset	Kernel	$ICOMP_{PERF}$	Training Err. CI	Testing Err. CI	GA. Freq.
{1, 2}	Laplace	-77.78	[0.00, 0.00]	[0.72, 0.72]	86
{1, 2}	Exponential	-71.81	[0.00, 0.00]	[0.57, 0.89]	10
{1, 2}	Inv. Multi-Quadric	-55.2	[0.00, 0.00]	[0.33, 1.29]	1
{1}	Laplace	9.12	[0.00, 2.62]	[0.45, 2.33]	3

Not only was the best solution selected by 86% of the replications, more than half of the total generations selected it. Using the Laplace kernel function on both variables of this data is a very clear winner with a 0.72% probability of misclassification, not that any of these four models had unreasonable error rates. Figure 2 has the scatter plots plus the contours for the top two solutions. Note the clear separation boundary between each group - hence the low misclassification rates are not unexpected. Both models used a small number of support vectors - 14 and 11, respectively. Less than 10% of the observations was needed, which shows the efficiency of support vector classification. Clearly the full GA with cross-validation model was overkill for this dataset, as complete enumeration would only need to evaluate 27 solutions. It has been included here purely for the visualization the support-vector-based separation areas, which will not be possible for the other examples.



(a) With Laplace Kernel (b) With Exponential Kernel

Figure 2: Demonstrating Separation with the Two Best KDA Models. The classes are marked by "x" and "+", and "**o**" indicates the support vectors.

4.2. Wine Composition Data

Our next example is the wine recognition dataset of M. Fiorina, *et al.*, used in [1]. These data are the results of a chemical analysis of $n = 178$ wines grown in the same region in Italy but derived from $K = 3$ different cultivars ($n_1 = 59$, $n_2 = 71$, $n_3 = 48$). The analysis determined the quantities of $p = 13$ chemical constituents found in each of the three types of wines. The variables are shown in Table 3.

Table 3: Wine Data Variables.

Variable		Variable	
x_1	Alcohol	x_8	Non-flavonoid Phenols
x_2	Malic Acid	x_9	Proanthocyanins
x_3	Ash	x_{10}	Color Intensity
x_4	Alkalinity of Ash	x_{11}	Hue
x_5	Magnesium	x_{12}	OD280/OD315 of Diluted Wines
x_6	Total Phenols	x_{13}	Proline
x_7	Total Flavonoids		

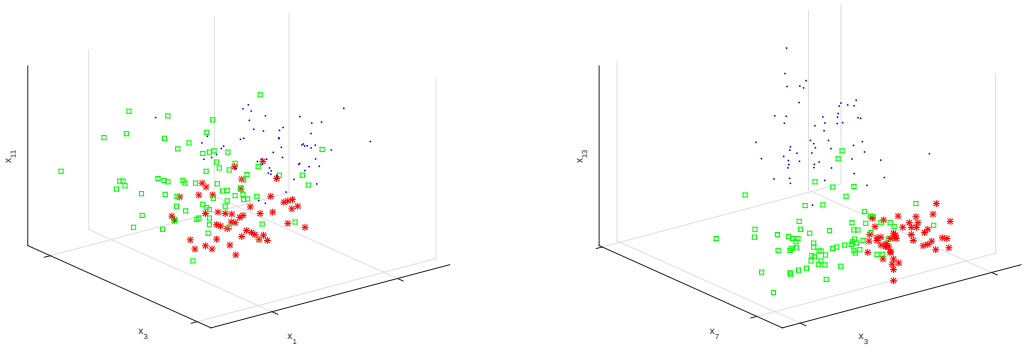
With this dataset, there are $9 \times \left(2^{13} - 1\right) = 73,719$ possible nontrivial solutions; the CVGA with 100 replications of the GA explored nearly half of the solution space. None of the generations selected the saturated model with all features as the best; the most frequently selected kernel functions were the Quadratic (25%) and Exponential (18%). In Table 4, we report the best 6 models with their classification results. The top 6, as opposed to a round number, were chosen, since they had identically perfect training errors.

Table 4: Top 6 Models Selected for Wine Data.

Subset	Kernel	$ICOMP_{PERF}$	Training Error	Testing Error
$\{1-4,7,8,11-13\}$	Quadratic	-229.61	0.00%	0.70%
$\{1,3,7,11,13\}$	Cubic	-222.45	0.00%	0.70%
$\{1,3,6-8,11-13\}$	Quadratic	-145.87	0.00%	1.41%
$\{1,3,4,7,8,9,11,12\}$	Quadratic	-127.64	0.00%	1.41%
$\{1-3,7,10,13\}$	Quadratic	-125.01	0.00%	1.41%
$\{1,4,6,9-13\}$	Quadratic	-123.36	0.00%	1.41%

All the subset models shown here exhibit substantial overlap - almost all include at least the first, third, and thirteenth features. In considering the classification errors, there's no difference between the best two models. Additionally, since the $ICOMP_{PERF}$ scores are so close, it's likely, due to the random variation from cross validation, that the scores are not significantly different. However, the first model uses nine variables, while the second, which achieves the same low error, uses only five. Hence, the **principle of parsimony drives us to prefer the subset** $\{1,3,7,11,13\}$.

In Figure 3, we show 3-way scatterplots for two of the possible ten combinations of these five variables. The plots selected were fairly representative of all 10 - there were no 3-way plots with clear separation of all classes *in the original data space*. This suggests that the kernel methods boosted classification performance as we expect.



(a) Features $1 \times 3 \times 11$ (b) Features $3 \times 7 \times 13$

Figure 3: Trivariate Grouped Scatterplots for the Best Wine Subset.

4.3. Aorta Nuclear Resonance Image Data

Our final example is by application to medical imaging data from a study of heart tissue. Hardening of the arteries is a leading cause of death and debility in the industrial world. In the U.S. alone 13 million Americans suffer from heart attacks, and $90,000$ people die from heart disease annually. Nuclear Magnetic Resonance (NMR) imaging has been used to aid clinical identification of fatty tissues in the arteries to aid in early detection of heart attacks. The aorta data analyzed here was collected by Pearlman [13] at the Medical School of the University of Virginia. There are observations from $n = 418$ patients on 16 different image acquisition variables. Including direction and orientation variables, we have $p = 20$ variables. The first $n_1 = 194$ patients exhibited *early atheroma*, and the remaining $n_2 = 224$ patients were clinically deemed *healthy*.

For this dataset which exhibits marked non-normality, there are $9,437,175$ possible subset+kernel combinations. The CVGA ran 100 replications of the genetic algorithm, with each running for twenty generations with a population size of twenty-five. With the elitism rule turned on, our modeling process evaluated **at most** $69,000$ unique solutions - 0.73% of the solution space.

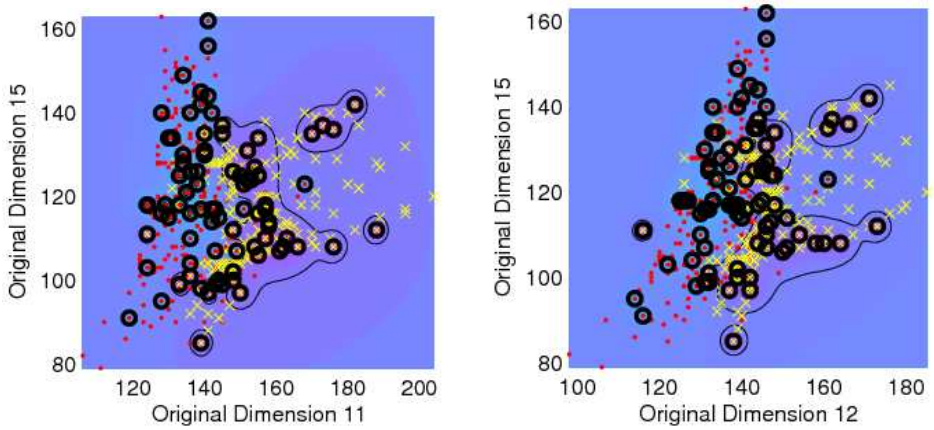
When sorted by the $ICOMP_{PERF}$ score, the top forty solutions used the Cauchy kernel, had indistinguishable scores, with identical testing error rates of 0.30% and varying subsets of features. None of the generations selected the fully saturated model. Among the indistinguishable (by score and testing error) solutions, the only kernels selected were the Cauchy and Inverse Multi-Quadric kernels.

Before running the full modeling procedure, we performed some exploratory data analysis using the full set of features. In this analysis, we generated 100 training / testing cross-validation samples, and fit each of the 9 kernels to all. With the full dataset, the 95% confidence interval of the testing error rates with the Cauchy was abysmal: $[43.80\%, 47.65\%]$. This once again demonstrates the clear benefit of optimal feature selection.

Table 5 show five of the best models. While the Cauchy kernel appears superior for this data, it seems there is no clear best set of features, since all of the top five performed nearly perfectly. The top model with the Cauchy kernel has dramatically reduced the dimensionality of the data, from $p = 20$ variables down to $p* = 6$. Figure 4 displays the poor separation in the original data space for two sets of pairs of the original features.

Table 5: 5 of the Best Models Selected for Aorta Data.

Subset	Kernel	$ICOMP_{PERF}$	Training Error	Testing Error
$\{5, 7, 8, 11, 12, 15\}$	Cauchy	-530.05	0.00%	0.30%
$\{4, 5, 8, 9, 11, 16, 18\}$	Cauchy	-530.05	0.00%	0.30%
$\{4, 7-9, 11-13, 15, 18\}$	Linear	-440.22	0.00%	0.30%
$\{3, -5, 7, 9, 12-16, 19\}$	Cubic	-145.87	0.00%	0.30%
$\{2-7, 9, 10, 12, 14, 15, 17, 19\}$	Homogenous Poly	-127.64	0.00%	0.30%



(a) Features 11×15 (b) Features 12×15

Figure 4: Bivariate Grouped Scatterplots for Aorta Data Showing Poor Separation.

5. Concluding Remarks

In summary, we have developed and applied a new hybrid stochastic algorithm, CVGA, which combines the strengths of cross-validation and the genetic algorithm. Within the GA, we've used a novel encoding strategy to create chromosomes which simultaneously encode for feature selection and kernel selection. We've applied the CVGA to regularized support vector

classification with kernel discriminant analysis. However, CVGA can be useful for any machine learning problem that requires cross-validation and exploration of a large solution space. The only requirement is the ability to encode solutions as GA chromosomes.

Recall that we set the kernel function parameters to certain values *a priori*, instead of estimating them from the data. This research could be further extended to allow estimation of the parameters. One approach might be to merge the CVGA with the data-adaptive technique presented in [12].

References

[1] S Aeberhard, D Coomans, and O de vel. Comparison of Classifiers in High Dimensional Settings. Technical Report 92-02, Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, 1992.

[2] N Aronszajn. Theory of Reproducing Kernels. In *Transactions of the American Mathematical Society*, volume 68, pages 337–404, 1950.

[3] H Bozdogan. ICOMP: A New Model-Selection Criteria. In H H Bock, editor, *Classification and Related Methods of Data Analysis*, pages 599–608. Elsevier Science Publishers, Amsterdam, The Netherlands, 1988.

[4] H Bozdogan. On the Information-Based Measure of Covariance Complexity and its Application to the Evaluation of Multivariate Linear Models. *Communication in Statistics, Theory and Methods*, 19:221–278, 1990.

[5] H Bozdogan. Akaike's Information Criterion and Recent Developments in Information Complexity. *Journal of Mathematical Psychology*, 44:62–91, March 2000.

[6] M Chen. Estimation of Covariance Matrices Under a Quadratic Loss Function. Research Report S-46, Department of Mathematics, SUNY at Albany, 1976.

[7] D Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Col, Inc.,, Boston, USA, 1989.

[8] R Haupt and S Haupt. *Practical genetic algorithms*. John Wiley, Hoboken, USA, 2004.

[9] J H Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The University of Michigan Press, Ann Arbor, USA, 1975.

[10] J H Holland. Genetic Algorithms. *Scientific American*, 267:66–72, 1992.

[11] C Hsu and C Lin. A comparison of methods for multiclass support vector machines. In *IEEE Transactions on Neural Networks*, volume 13, pages 415–425, 2002.

[12] C Liberati, J A Howe, and H Bozdogan. Data Adaptive Simultaneous Parameter and Kernel Selection in Kernel Discriminant Analysis Using Information Complexity. *Journal of Pattern Recognition Research*, 4(1):119–132, 2009.

[13] J Pearlman. *Nuclear Magnetic Resonance Spectral Signatures of Liquid Crystals in Human Atheroma As Basis For Multi-Dimensional Digital Imaging of Atherosclerosis*. PhD thesis, University of Virginia, 1986.

[14] S Press. Estimation of a Normal Covariance Matrix. Technical report, University of British Columbia, 1975.

[15] C Thomaz. *Maximum Entropy Covariance Estimate for Statistical Pattern Recognition*. PhD thesis, University of London and Diploma of the Imperial College (D.I.C.), 2004.