Robust Margin Based Classifiers For Small Sample Data

by

Sidharth Gupta

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2011 by the
Graduate Supervisory Committee:

Seungchan Kim, Chair
Bruno Welfert
Baoxin Li

ARIZONA STATE UNIVERSITY

May 2011

ABSTRACT

In many classification problems data samples cannot be collected easily, example in drug trials, biological experiments and study on cancer patients. In many situations the data set size is small and there are many outliers. When classifying such data, example cancer vs normal patients the consequences of mis-classification are probably more important than any other data type, because the data point could be a cancer patient or the classification decision could help determine what gene might be over expressed and perhaps a cause of cancer. These mis-classifications are typically higher in the presence of outlier data points. The aim of this thesis is to develop a maximum margin classifier that is suited to address the lack of robustness of discriminant based classifiers (like the Support Vector Machine (SVM)) to noise and outliers. The underlying notion is to adopt and develop a natural loss function that is more robust to outliers and more representative of the true loss function of the data. It is demonstrated experimentally that SVM's are indeed susceptible to outliers and that the new classifier developed, here coined as Robust-SVM (RSVM), is superior to all studied classifier on the synthetic datasets. It is superior to the SVM in both the synthetic and experimental data from biomedical studies and is competent to a classifier derived on similar lines when real life data examples are considered.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Robust-Support Vector Machine, a robust solution to outliers

It is known that support vector machines and learning models like discriminant classifiers, optimize a loss function that is typically a hinge loss or its variant. These learning models face the issue of lack of robustness to noise and outliers [25], which is addressed in this thesis. By adopting a loss function that represents the true nature of the loss, rather than an analytically simple one it is shown that the new classifier can compete with contemporary ones like the SVM. This thesis is motivated from previous work[11]. But while the previous work extended the Linear discriminant Classifier(LDA), the RSVM extends Primal Support Vector Machines[3].

This chapter will discuss first the contemporary loss functions used in maximum-margin classifiers and their inherent limitations, its limitations are an important motivation for the RSVM. Strong arguments for the choice of its analytical design are also presented. And finally the organization of the rest of the thesis is laid out.

SVM and its limitations

Given a training set $\{(\mathbf{x_i}, y_i)\}_{1 \leq i \leq n}$, $\mathbf{x_i} \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$ recall that the primal SVM optimization problem is usually written as:

$$\min_{w,b} ||\mathbf{w}||^2 + C \sum_{i=1}^{n} \xi_i^p \quad (1)$$

$$\textit{under the constraints} :$$

$$y_i(\mathbf{w}.\mathbf{x_i^t} + b) \geq 1 - \xi_i, \quad \xi_i \quad \geq 0$$

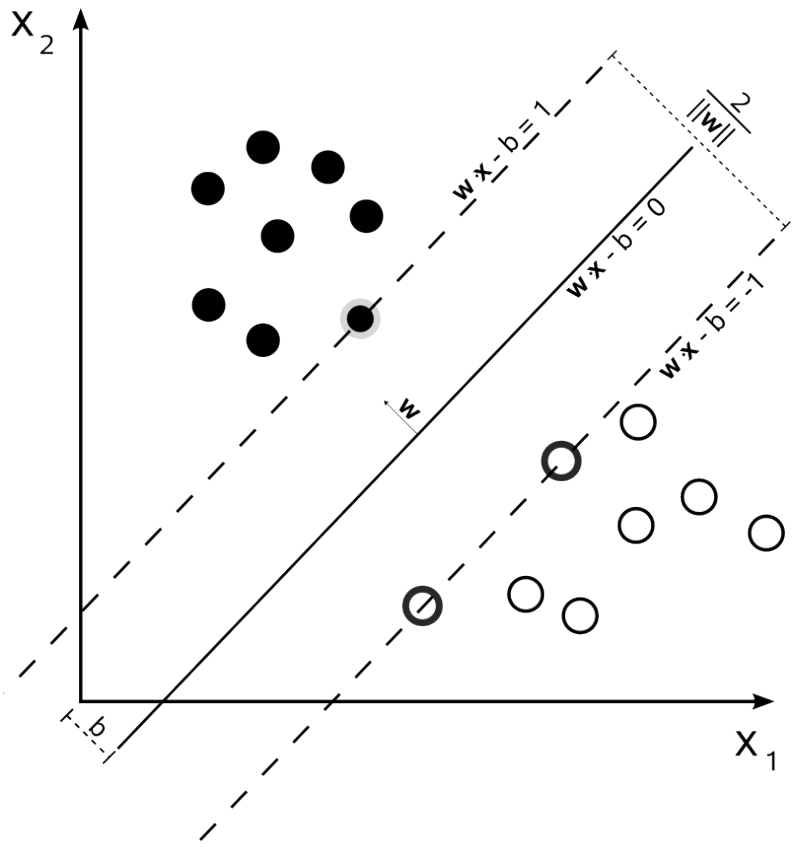

Figure 1.1: Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

where p is either 1 (hinge loss) or 2 (quadratic loss) and $< w, b >$ is a weight vector that represents the separating hyperplane that is used to classify the data. The intuitive idea behind the SVM is that we want to choose $< w, b >$ to maximize the margin, or distance between the parallel hyperplanes that are as far apart as possible while still separating the data. The geometrical depiction is shown in figure (1.1). The two separating hyperplanes straddling the data can be described by the equations:

$$\mathbf{w}.\mathbf{x_i^t} + b = 1$$

$$and \quad \mathbf{w}.\mathbf{x}_i^t + b = -1$$

By using geometry, we find the distance between these two hyperplanes is $\frac{2}{||\mathbf{w}||}$, and our objective is to minimize $||\mathbf{w}||^2$ (the square is added to get rid of the

root). The problem reduces to expression 1 once the constraints are taken into consideration. Note that this is a quadratic optimization problem and depending on the value of p at this point, in the existent literature, typically there are usually two main methodologies followed to solve this problem i.e. either in the primal(used for p=2) or in the dual(used mainly for p=1). Both the solutions also exploit the kernel trick to deal with linearly inseparable data. The SVM is known to be less prone to the curse of dimensionality and provides a superior solution to other classifier methodologies like neural networks, logistic regression etc.

However it has been observed that in the small sample setting the dimensionality of the data, the complexity of the kernel function and projection into higher dimensional space can introduce problems of over fitting as observed in[17, 19]. In experimental biological data like micro-array data it is always the case that genes are much more than the number of samples so typically some dimensionality reduction technique is always applied. It is worth noting that recent developments especially in neighborhood embeddings in low dimensional spaces like [20, 7] are worth looking at and are state of the art for dimensionality reduction and their importance in analyzing experimental biological data cannot be overstated.

More importantly it is well known that outlier robustness of SVM's is an issue and the experiment, shown illustratively, in Figure 1.2 and [25] have found that the solution for the soft margin SVM using the hinge loss is plagued by outliers, that bear a maximal effect on the optimal solution. The problems of over-fitting in a small sample setting and the outlier robustness issue are strong motivations to improve upon the SVM. Next the choice of a primal over a dual solution is argued for the design of the RSVM.

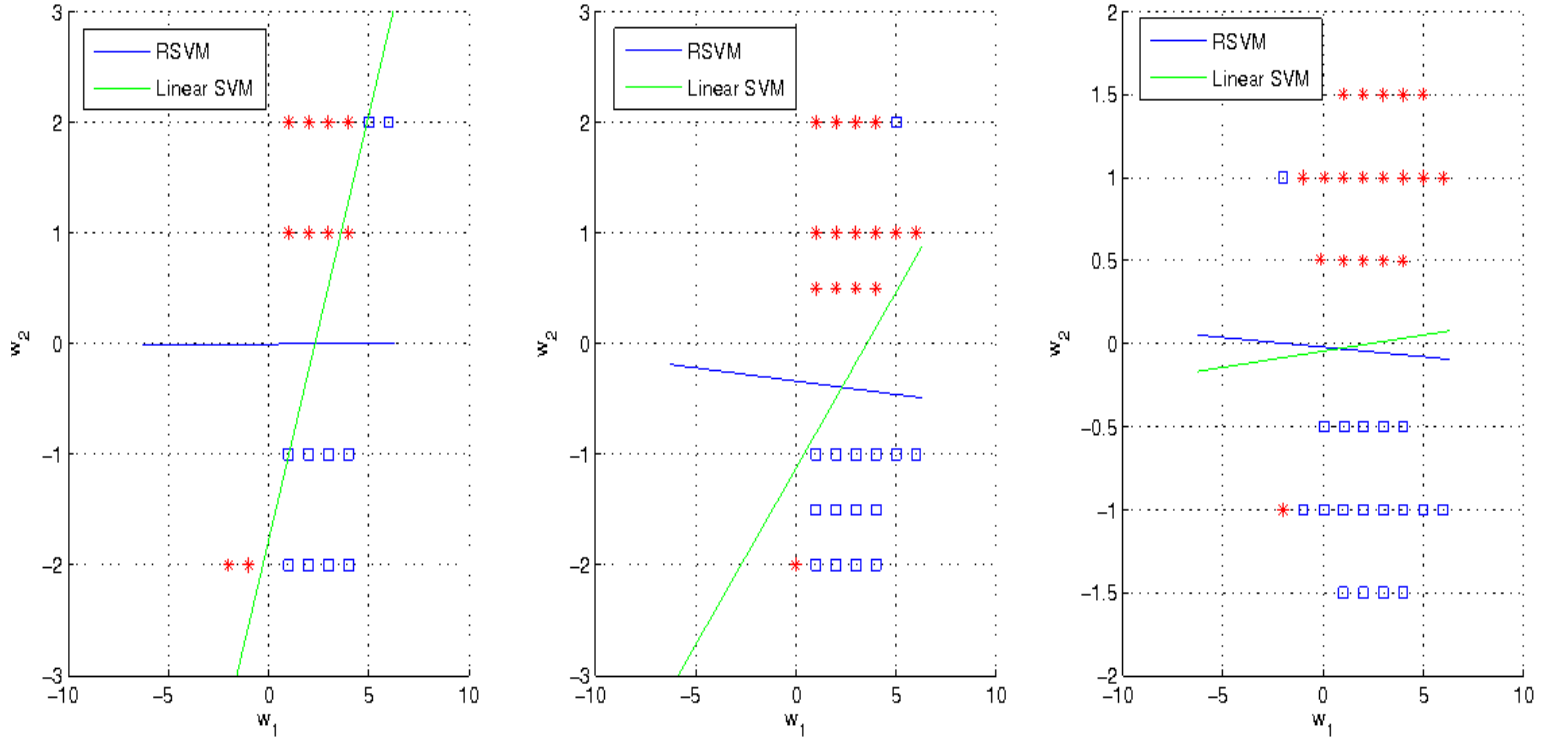

Figure 1.2: A experimental result showing the margin constructed by the SVM and the RSVM. Showing that outliers have a measurable effect on margin formation. The margin corrects itself only after more points are added to the data.

A primal solution for the RSVM

It is shown that the SVM optimization problem is solvable in the primal and it is also shown that when deriving an approximate solution to the SVM, the optimization in the primal is superior to the that in the dual[3]. The time and space complexities of solving the primal and the dual have been shown, analytically, to be the same for the primal and the dual forms; In fact when it comes to an approximate solution, primal optimization is superior because it is more focused on minimizing what we are interested in: the primal objective function rather than its dual[3]. Further it is proved in Appendix A that the primal and dual solutions of the SVM recover essentially the same solution. This is the main motivation for a primal solution to design the RSVM.

Previous work

The issue of the SVM with outliers is demonstrated in the previous sections and a motivation for a primal classifier has been discussed, but what is the solution that we should adopt? It is already shown that it is possible to improve the error estimates by considering the sample spread of the data points as a measure for reliable error estimation[11]. Intuitively put this is analogous to drawing a ball around the data points and then minimizing the volume of the ball that is cut by the hyperplane. Since the area represents a spread, its minimization automatically reveals a loss function also. But here in lies a problem; In higher dimensional space this problem is very hard to solve as the probability mass tends to be concentrated in a thin shell at a finite radius of a D-Dimensional ball[5]. The loss function becomes hard to solve analytically, thus some simplification is required. A solution for this was proposed in [11] wherein the loss function chosen for RSVM is a simple circular Gaussian, it is in fact the Error Function, with equal variance along the diagonal. The RSVM is also designed on similar lines. The

variance is estimated using the data distribution, similar to[11]. The algorithm is thus parameterized by the variance of the data and is not a hyperparameter. It still retains the regularizer from the Primal SVM as an hyperparameter. The nature of the loss function makes the design unique and different from those included in the current thesis.

Organization of the thesis

The margin formulation is mathematically intensive and is presented in chapter 2 of the paper. Chapter 2 will also define the loss function and discuss its mathematical properties. Gradient and second order based optimization techniques are exploited to derive an analytical form for a new margin based classifier. The loss function, based on sample spread and the max margin definition, reveals a convex optimization function solved in chapter 2. The loss function proposed is naturally convex, albeit mathematically complex. The focus of the paper, for now, will be more on providing a strong proof of concept and laying down the basic ground work for a more robust classifier rather than performance. Experiments on some data sets are done on chapter 3. First the performance of the RSVM is compared against the Linear SVM and Sigma Classifier on synthetic dataset and then it is compared to case of the real life datasets also.

Chapter 2

MATHEMATICAL FORMULATION

In this chapter first, the minimization function along with the new loss function is outlined. The method to solve this optimization function and its mathematical characteristics are discussed next. Then some notation is mentioned, followed by the mathematical derivation of the classifier in section 2.4 and 2.5. Finally the numerical and computational aspects are discussed in the last section.

The objective function and its solution

The R-SVM objective function that needs to be optimized is defined as the expression:

$$\lambda||w||^2 + \sum_{i \in n} L_g(\mathbf{w}, \mathbf{x}_i, b) \quad (\#1.0)$$

The normalized Gaussian with zero mean is defined to be:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{x^2}{2\sigma^2})$$

Now the error L_g is defined as:

$$L_g(\mathbf{w}, \mathbf{x}_i, b) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma_{c_k}} \int_{d_i}^{\infty} exp(\frac{-x^2}{2\sigma_{c_k}^2}) dx & 0 < y_i(\mathbf{w}.\mathbf{x}_i^t + b) \leq 1 \\ \frac{1}{\sqrt{2\pi}\sigma_{c_k}} \int_{-\infty}^{d_i} exp(\frac{-x^2}{2\sigma_{c_k}^2}) dx & y_i(\mathbf{w}.\mathbf{x}_i^t + b) < 0 \\ 0 & otherwise \end{cases} \quad (\#1.1)$$

where d_i is the function output of $d_i(\mathbf{w}, \mathbf{x}_i, b) = \frac{|\mathbf{w}.\mathbf{x}_i^t + b|}{||\mathbf{w}||}$ which is the euclidean distance of the point from the hyper-plane described by $< \mathbf{w}, b >$. The first and second derivatives of the system will be solved analytically. $\sigma_{c_k}$are the

class specific parameters (not hyper-parameter) derived from the spreading of the data distribution. The loss function is similar in shape to the cumulative distribution curve of the exponential distributions of form: $p(x) = exp(-x^2)$ and it may give the reader the impression that the sigmoid function is similar in shape to this and may be used as a substitute. But this is not true because the error bounds for the sigmoid are not the same as the Gaussian Error Function.

The following section will present a closed form solution of the first and the second derivatives of the loss function. The weights can be updated using newton's method:

$$\mathbf{w} = \mathbf{w} + \eta H^{-1} \nabla \quad (\#1.2)$$

Section 2.3 is the first derivative evaluation. Section 2.4 is the evaluation of the Hessian (Second Derivative).

$$b = b + \eta \frac{dL_g/db}{d^2 L_g/db^2} \quad (\#1.3)$$

The choice of a Gaussian as a loss function is reasonable as a lot of random data phenomenon are considered Gaussian in nature and a lot of unsupervised and supervised learning algorithms like the Gaussian Mixture Models and PCA depend on it[18, 24]. Certainly an analysis of the data and previous studies on similar data should be studied before hand to ascertain the nature of the data and applicability of certain method.

Another important reason why the integral of a circular Gaussian was chosen as the loss function is because has a closed mathematical form represented by 1.1 and more pertinently its derivatives are also having a closed form. Other distributions like Poisson or chi-square were found to be worth studying but there

derivatives did not have simple closed form and thus harder to handle analytically and analyze.

Convexity, smoothness of the loss function

First lets look at the error function itself, $erfc$. Its convexity will in turn prove the convexity of the loss function. Let the function be convex

$$\begin{aligned} erfc(t*x_1+(1-t)*x_2) &\leqslant t*erfc(x_1)+(1-t)*erfc(x_2) \\ \int_{tx_1+(1-t)x_2}^{\infty} exp(-x^2)dx &\leqslant t*\int_{x_1}^{\infty} exp(-x^2)dx+(1-t)\int_{x_2}^{\infty} exp(-x^2)dx \end{aligned}$$

where $t \in [0,1]$ also $x_1 \leq t*x_1+(1-t)*x_2 \leq x_2$. Thus one can split the L.H.S. of the above and rearrange the R.H.S. as:

$$\begin{aligned} \int_{t*x_1+(1-t)*x_2}^{x_2} exp(-x^2)dx+\int_{x_2}^{\infty} exp(-x^2)dx &\leqslant t\left(\int_{x_1}^{\infty} exp(-x^2)dx-\int_{x_2}^{\infty} exp(-x^2)dx\right) \\ &\quad +\int_{x_2}^{\infty} exp(-x^2)dx \\ \int_{t*x_1+(1-t)*x_2}^{x_2} exp(-x^2)dx &\leqslant t*\int_{x_1}^{x_2} exp(-x^2)dx \end{aligned}$$

Which is true for any t. The convexity is not strict. As the area under the Gaussian is unity. The error function is upper bounded. It is naturally smooth according to the definition 1.1. In contrast the hinge loss is naturally not smooth and is not a proper scoring function[13]. It is employed largely due to its mathematical simplicity and because it gets the mathematical sign right and is

able to approximate the $0-1$ error. There is a great debate within the machine learning community about weather to use real life loss function that the data may represent or those that are mathematically simple. As it is demonstrated by the better experimental results and in[11], selecting a natural loss function can give better results. The mathematically correct idea is to select a loss function that is an approximation of the $0-1$ loss[22]which is satisfied in this case because the maximum error contributed by a point is unity (area under Gaussian) in the limiting case.

Notation

Bold face small letters will denote vectors. Capital letters are representative of Matrices. Thus the expression$\mathbf{x} * z$ has $\mathbf{x}$ as a vector and z as a scalar, it is the same as multiplying each component of $\mathbf{x}$ by a scalar z. $'.'$ is generally used to represent the dot product between vectors. $\otimes$ is the outer product of vectors or the Kronecker product. ".$*$" will represent point wise multiplication. Each vector is considered to be a row vector.$\mathbf{x}^t$ is the transpose of a vector.

The Gradient

Now the derivative of the error function definition in (#1.1) will be defined by the Leibniz rule:

$$\frac{d}{dx}\int_{f_1(x)}^{f_2(x)} g(t)\,dt = g(f_2(x)).f_2'(x) - g(f_1(x)).f_1'(x)$$

Note that the derivatives corresponding to the upper limit ($\pm\infty$) will disappear and the derivative becomes:

$$\frac{\partial}{\partial w}L_g(\mathbf{w}, b, \mathbf{x}_i)$$

$$= \begin{cases} \frac{1}{\sqrt{2\pi}\sigma_{c_k}}\frac{\partial}{\partial w}\int_{d_i}^{\infty} exp(\frac{-x^2}{2\sigma_{c_k}^2})dx & 0 < y_i(\mathbf{w}.\mathbf{x}_i^t + b) \leqslant 1 \\ -\frac{1}{\sqrt{2\pi}\sigma_{c_k}}\frac{\partial}{\partial w}\int_{d_i}^{-\infty} exp\left(-\frac{x^2}{2\sigma_{ck}^2}\right)dx & y_i(\mathbf{w}.\mathbf{x}_i^t + b) < 0 \end{cases} \quad (\#1.4)$$

Before differentiating. Put $\frac{x}{\sqrt{2}\sigma_{ck}} = y$. Thus $dy = \frac{dx}{\sqrt{2}\sigma_{ck}}$. The lower limit can be redefined as $y = \frac{d_i(\mathbf{w}.,\mathbf{x},b)}{\sqrt{2}\sigma_{ck}} = \frac{|\mathbf{w}.\mathbf{x^t}_i+b|}{||\mathbf{w}||\sqrt{2}\sigma_{ck}}$. Thus above becomes:

$$\frac{\partial}{\partial w}L_g(\mathbf{w}, b, \mathbf{x}_i)$$

$$= \begin{cases} \frac{1}{\sqrt{\pi}}\frac{\partial}{\partial w}\int_{\frac{d_i}{\sqrt{2}\sigma_{ck}}}^{\infty} exp(-x^2)dx & 0 < y_i(\mathbf{w}.\mathbf{x}_i^t + b) \leq 1 \\ -\frac{1}{\sqrt{\pi}}\frac{\partial}{\partial w}\int_{\frac{d_i}{\sqrt{2}\sigma_{ck}}}^{-\infty} exp\left(-x^2\right)dx & y_i(\mathbf{w}.\mathbf{x}_i^t + b) < 0 \end{cases} \quad (\#1.5)$$

Thus the derivative described in terms of the Gaussian Error Function, erfc is:

$$\frac{\partial}{\partial w}L_g(\mathbf{w}, b, \mathbf{x}_i) =$$

$$\begin{cases} \frac{\partial}{\partial w}\left(\frac{erfc(\frac{d_i}{\sqrt{2}\sigma_{ck}})}{2}\right) & 0 < y_i(\mathbf{w}.\mathbf{x}_i^t + b) \leqslant 1 \\ \frac{\partial}{\partial w}\left(\frac{1}{2} + \frac{erf\left(\frac{d_i}{\sqrt{2}\sigma_{ck}}\right)}{2}\right) & y_i(\mathbf{w}.\mathbf{x}_i^t + b) < 0 \end{cases} \quad (\#1.6)$$

Note that the value of the partial derivatives in (1.5) is exactly the same with respect to the $\pm\infty$ upper-limits:

$$\frac{\partial}{\partial \mathbf{w}}\left(\int_{\frac{d_i}{\sqrt{2}\sigma_{ck}}}^{\pm\infty} exp\left(-x^2\right)dx\right) = -\frac{1}{\sqrt{2}\sigma_{ck}} * exp\left(-d_i^2(\mathbf{w},\mathbf{x_i},b)\right) * \frac{\partial(d_i(\mathbf{w},\mathbf{x}_i,b))}{\partial \mathbf{w}}$$

It is also important to note that since $|..|$ function in d_i is not differentiable one need to remove the discontinuity by defining two functions for d_i, as done in (1.4). The derivative is defined at all points except where $\mathbf{w}.\mathbf{x}_i + b = 0$.

$$\begin{aligned} &= -\frac{1}{\sqrt{2}\sigma_{ck}} * exp\left(-\left(\frac{d_i(\mathbf{w},\mathbf{x}_i,b)}{\sqrt{2}\sigma_{ck}}\right)^2\right) * \frac{\partial(\frac{\mathbf{w}.\mathbf{x}_i^t+b}{||\mathbf{w}||})}{\partial \mathbf{w}} \\ &= -\frac{1}{\sqrt{2}\sigma_{ck}} * exp\left(-\left(\frac{d_i(\mathbf{w},\mathbf{x}_i,b)}{\sqrt{2}\sigma_{ck}}\right)^2\right) * \left(\frac{\mathbf{x}_i^t}{||w||} - \frac{\mathbf{w}^t}{||w||^3}.*(\mathbf{w}.\mathbf{x}_i^t+b)\right) \end{aligned}$$

The exponential term indicates that the derivative is directly dependent on distance of the data from the hyperplane.

Note that the sum error of the system from equation 1.0 is:

$$E = \sum_{i \in n_{sv}} L_g(w,b,x_i)$$

Thus the derivative, written with the summation taken into account is:

$$\begin{aligned} &= -\sum_{i\in n_{sv}} \frac{1}{\sqrt{2}\sigma_{ck}} * exp\left(-\left(\frac{d_i(\mathbf{w},\mathbf{x}_i,b)}{\sqrt{2}\sigma_{ck}}\right)^2\right) * \left(\frac{\mathbf{x}_i^t}{||w||} - \frac{\mathbf{w}^t}{||w||^3} * (\mathbf{w}.\mathbf{x}_i^t+b)\right) \\ &= -\sum_{i\in n_{sv}} \frac{1}{\sqrt{2}\sigma_{ck}} exp\left(-\left(\frac{d_i(\mathbf{w},\mathbf{x}_i,b)}{\sqrt{2}\sigma_{ck}}\right)^2\right) * \frac{\mathbf{x}_i^t}{||\mathbf{w}||} - \frac{exp\left(-\left(\frac{d_i(\mathbf{w},\mathbf{x}_i,b)}{\sqrt{2}\sigma_{ck}}\right)^2\right)}{||\mathbf{w}||} * \\ &\quad \frac{\mathbf{w}^t}{||\mathbf{w}||} * \frac{(\mathbf{w}.\mathbf{x}_i^t+b)}{||\mathbf{w}||} \\ &= -\sum_{i\in n_{sv}} \frac{1}{\sqrt{2}\sigma_{ck}} * f_i * \frac{\mathbf{x}_i^t}{||\mathbf{w}||} - \frac{f_i}{||\mathbf{w}||} * g_i * \frac{\mathbf{w}^t}{||\mathbf{w}||} \end{aligned}$$

Combining with 2 above and the derivative of the first term of (1.0), viz $\lambda||w||^2$ is:

$$\nabla = \begin{cases} 2\lambda \mathbf{w}^t - \frac{1}{\sqrt{2\pi}\sigma_{ck}} \sum_{i \in n_{sv}} f_i * \frac{\mathbf{x}_i^t}{||\mathbf{w}||} - \frac{f_i}{||\mathbf{w}||} * g_i * \frac{\mathbf{w^t}}{||\mathbf{w}||} & 0 < y_i(\mathbf{w}.\mathbf{x}_i^t + b) \leqslant 1 \\ 2\lambda \mathbf{w}^t + \frac{1}{\sqrt{2\pi}\sigma_{ck}} \sum_{i \in n_{sv}} f_i * \frac{\mathbf{x}_i^t}{||\mathbf{w}||} - \frac{f_i}{||\mathbf{w}||} * g_i * \frac{\mathbf{w^t}}{||\mathbf{w}||} & y_i(\mathbf{w}.\mathbf{x}_i^t + b) < 0 \end{cases} \quad (\#1.7)$$

Note that the gradient is a column vector. Where
$f_i = f(\mathbf{w}, \mathbf{x_i}, b) = exp\left(-\left(\frac{d_i(\mathbf{w},\mathbf{x}_i,b)}{\sqrt{2}\sigma_{ck}}\right)^2\right)$ and $g_i = g(\mathbf{w}, \mathbf{x_i}, b) = \frac{\mathbf{w}.\mathbf{x}_i^t + b}{||\mathbf{w}||}$.
Note that these are scalars and defined for a point i in the data.

Also the partial derivatives of these quantities are defined as follows:

$$\begin{aligned} \mathbf{g_i'} = \frac{\partial g_i}{\partial \mathbf{w}} &= \frac{\partial \left(\frac{\mathbf{w}.\mathbf{x}_i^t + b}{||\mathbf{w}||}\right)}{\partial \mathbf{w}} \\ &= \frac{||\mathbf{w}|| * \frac{\partial}{\partial \mathbf{w}}(\mathbf{w}.\mathbf{x}_i^t + b) - \frac{\mathbf{w}^t}{||\mathbf{w}||} * (\mathbf{w}.\mathbf{x}_i^t + b)}{||\mathbf{w}||^{\mathbf{2}}} \\ &= \left(\frac{\mathbf{x}_i^t}{||\mathbf{w}||} - \frac{\mathbf{w}^t}{||\mathbf{w}||^3} * (\mathbf{w}.\mathbf{x}_i^t + b)\right) \end{aligned}$$

This is a vector and also the derivative $\mathbf{f_i'}$:

$$\begin{aligned} \mathbf{f_i'} = \frac{\partial f}{\partial \mathbf{w}} &= \frac{\partial\, exp\left(-\left(\frac{\mathbf{w}.\mathbf{x}_i^t + b}{||\mathbf{w}||\sqrt{2}\sigma_{ck}}\right)^2\right)}{\partial \mathbf{w}} \\ &= -exp\left(-\left(\frac{\mathbf{w}.\mathbf{x}_i^t + b}{||\mathbf{w}||\sqrt{2}\sigma_{ck}}\right)^2\right) * \frac{\partial \left(\frac{\mathbf{w}.\mathbf{x}_i^t + b}{||\mathbf{w}||\sqrt{2}\sigma_{ck}}\right)^2}{\partial w} \end{aligned}$$

$$\begin{aligned} &= -exp\left(-\left(\frac{\mathbf{w}.\mathbf{x}_i^t + b}{||\mathbf{w}||\sqrt{2}\sigma_{ck}}\right)^2\right) * \frac{\mathbf{w}.\mathbf{x}_i^t + b}{||\mathbf{w}||} * \frac{\partial \left(\frac{\mathbf{w}.\mathbf{x}_i^t + b}{||\mathbf{w}||}\right)}{\partial \mathbf{w}} \\ &= -\left(\frac{1}{2\sigma_{ck}^2} * f_i * g_i\right) * \frac{\partial g_i}{\partial \mathbf{w}} - \left(\frac{1}{2\sigma_{ck}^2} * f_i * g_i\right) .* \mathbf{g_i'} \end{aligned}$$

Both of these terms are vectors with the dimensionality of the data and are column vectors and will be used repeatedly in the computation of the Hessian (the second derivative).

The Hessian

Now the expression for second derivatives here will be taken over the two terms which is rewritten here for brevity:

$$\begin{aligned} & exp\left(-d_i^2(\mathbf{w},b,\mathbf{x}_i)\right) * \frac{\mathbf{x}_i^t}{||\mathbf{w}||} \ and \quad \mathbf{w}^t * exp\left(-d_i^2(\mathbf{w},b,\mathbf{x}_i)\right) * \frac{(\mathbf{w}.\mathbf{x}_i^t + b)}{||\mathbf{w}||^3} \\ = \ & \frac{f_i}{||\mathbf{w}||} * \mathbf{x}_i^t \ \ and \quad \left(\frac{f_i}{||\mathbf{w}||} * g_i\right) * \frac{\mathbf{w}^t}{||\mathbf{w}||} \quad respectively \end{aligned}$$

f_i, g_i are multivariate scalar valued functions defined in the last section. The second derivatives for these are solved, one term at a time. The first term, $\frac{f_i}{||\mathbf{w}||} * \mathbf{x}_i^t$ is a vector valued function[9]. The derivative of this is a matrix as:

$$\frac{\partial(f_i * \mathbf{x}_i^t)}{\partial \mathbf{w}} = \begin{bmatrix} \frac{x_i^1 \partial\left(\frac{f_i}{||\mathbf{w}||}\right)}{\partial \mathbf{w}} \\ \frac{x_i^2 \partial\left(\frac{f_i}{||\mathbf{w}||}\right)}{\partial \mathbf{w}} \\ \vdots \\ \frac{x_i^D \partial\left(\frac{f_i}{||\mathbf{w}||}\right)}{\partial \mathbf{w}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{x_i^1 \partial\left(\frac{f_i}{||\mathbf{w}||}\right)}{\partial w_1} & \frac{x_i^1 \partial\left(\frac{f_i}{||\mathbf{w}||}\right)}{\partial w_2} & \cdots & \frac{x_i^1 \partial\left(\frac{f_i}{||\mathbf{w}||}\right)}{\partial w_D} \\ \frac{x_i^2 \partial\left(\frac{f_i}{||\mathbf{w}||}\right)}{\partial w_1} & \frac{x_i^2 \partial\left(\frac{f_i}{||\mathbf{w}||}\right)}{\partial w_2} & \cdots & \frac{x_i^2 \partial\left(\frac{f_i}{||\mathbf{w}||}\right)}{\partial w_D} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{x_i^D \partial\left(\frac{f_i}{||\mathbf{w}||}\right)}{\partial w_1} & \frac{x_i^D \partial\left(\frac{f_i}{||\mathbf{w}||}\right)}{\partial w_2} & \cdots & \frac{x_i^D \partial\left(\frac{f_i}{||\mathbf{w}||}\right)}{\partial w_D} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{x_i^1 \partial h_i}{\partial w_1} & \frac{x_i^1 \partial h_i}{\partial w_2} & \cdots & \frac{x_i^1 \partial h_i}{\partial w_D} \\ \frac{x_i^2 \partial h_i}{\partial w_1} & \frac{x_i^2 \partial h_i}{\partial w_2} & \cdots & \frac{x_i^2 \partial h_i}{\partial w_D} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{x_i^D \partial h_i}{\partial w_1} & \frac{x_i^D \partial h_i}{\partial w_2} & \cdots & \frac{x_i^D \partial h_i}{\partial w_D} \end{bmatrix}$$

Thus it is the outer product $\mathbf{x}_i^t \otimes \frac{\partial h_i}{\partial \mathbf{w}}$. Where $h_i = \frac{f_i}{||\mathbf{w}||}$. Thus each row is defined as $\mathbf{h_i}' = \begin{bmatrix} \frac{x_i^D \partial h_i}{\partial w_1} \frac{x_i^D \partial h_i}{\partial w_2} \cdots \frac{x_i^D \partial h_i}{\partial w_D} \end{bmatrix} = \frac{\partial h_i}{\partial w}$

If we plug in the expression equivalent for f_i, one can evaluate $\mathbf{h_i'}$as:

$$\mathbf{h_i}' = \frac{||\mathbf{w}||\mathbf{f_i}' - \frac{\mathbf{w}}{||\mathbf{w}||} f_i}{||\mathbf{w}||^2} = \frac{\mathbf{f_i'}}{||\mathbf{w}||} - \frac{\mathbf{w}^t}{||\mathbf{w}||^3} * f_i$$

Note that this is a row vector.

Now for the derivative of the second term: $\mathbf{w^t} * exp\left(-d_i^2(\mathbf{w}, b, \mathbf{x_i})\right) * \frac{(\mathbf{w}.\mathbf{x}_i^t + b)}{||\mathbf{w}||^3}$. This is concisely written as $\frac{\mathbf{w}^t}{||\mathbf{w}||} * (h_i * g_i)$. The derivative of this can be written as:

$$\frac{\partial(\frac{\mathbf{w^t}}{||\mathbf{w}||}.*(h_i*g_i))}{\partial w} = \begin{bmatrix} \frac{\partial(\frac{w_1}{||\mathbf{w}||}*(h_i*g_i))}{\partial \mathbf{w}} \\ \frac{\partial(\frac{w_2}{||\mathbf{w}||}*(h_i*g_i))}{\partial \mathbf{w}} \\ \vdots \\ \frac{\partial(\frac{w_D}{||\mathbf{w}||}*(h_i*g_i))}{\partial \mathbf{w}} \end{bmatrix}$$

where each element expands into a row vector:

$$\begin{aligned} & \frac{\partial}{\partial \mathbf{w}}\left(\frac{w_j}{||\mathbf{w}||}*(h_i*g_i)\right) \\ = & \left[\frac{\partial}{\partial w_1}\left(\frac{w_j}{||\mathbf{w}||}*(h_i*g_i)\right), \frac{\partial}{\partial w_2}\left(\frac{w_j}{||\mathbf{w}||}*(h_i*g_i)\right), .. \right. \\ & \left. ...., \frac{\partial}{\partial w_D}\left(\frac{w_j}{||\mathbf{w}||}*(h_i*g_i)\right)\right] \\ = & \left[\frac{w_j w_1}{||\mathbf{w}||}*h_i*g_i + \frac{w_j}{||\mathbf{w}||}(h'_{i1}*g_i + g'_{i1}*h), \ \frac{w_j w_2}{||\mathbf{w}||}*h_i*g_i + \right. \\ & \frac{w_j}{||\mathbf{w}||}(h'_{i2}*g_i + g_{i2}*' h_i), ..., \\ & \left(\frac{1}{||\mathbf{w}||} - \frac{w_j^2}{||\mathbf{w}||^3}\right)*h_i*g_i + \frac{w_j}{||\mathbf{w}||}*(h'_{ij}*g_i + g'_{ij}*h_i), \\ & \left. ....., \frac{w_j w_D}{||\mathbf{w}||}*h_i*g_i + \frac{w_j}{||\mathbf{w}||}*(h'_{ij}*g_i + g'_{ij}*h_i)\right] \end{aligned}$$

Therefore

$$\frac{\partial(\frac{\mathbf{w^t}}{||\mathbf{w}||} * (h_i * g_i))}{\partial \mathbf{w}} =$$

$$\left[\frac{\mathbf{w^t} \otimes \mathbf{w}}{||\mathbf{w}||} * h_i * g_i + \frac{\mathbf{w^t} \otimes (\mathbf{h_i'} * g_i + \mathbf{g_i'} * h_i)}{||\mathbf{w}||} .* \begin{bmatrix} 0 & 1... & 1 \\ 1 & 0... & 1 \\ \vdots & \ddots & \vdots \\ 1 & 1... & 0 \end{bmatrix}_{DxD}\right.$$

$$\left. + \left(\left(\frac{\mathbf{1}}{||\mathbf{w}||} - \frac{\mathbf{w^2}}{||\mathbf{w}||^{\mathbf{3}}}\right) * h_i * g_i\right) \begin{bmatrix} 1 & 0... & 0 \\ 0 & 1... & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0... & 1 \end{bmatrix}_{DxD} \right]$$

Where $\otimes$ is the outer product or the Kronecker product and in the above is a matrix. The mathematics in the above is a little involved, It is instructive to work the above out on paper and to verify it, none of the steps of the derivation have been skipped due to complexity and for the sake of completeness.

Thus each element in the column matrix is a vector. Again one can simply plug in the values of f_i and g_i from the previous section. Thus finally writing the Hessian:

$$H = \begin{cases} I - \frac{1}{2\sqrt{2\pi}\sigma_{ck}} \sum_{i \in n_{sv}} \left[\begin{bmatrix} x_i^1 \partial\left(\frac{f_i}{||\mathbf{w}||}\right) \\ \hline \partial \mathbf{w} \\ x_i^2 \partial\left(\frac{f_i}{||\mathbf{w}||}\right) \\ \hline \partial \mathbf{w} \\ \vdots \\ x_i^D \partial\left(\frac{f_i}{||\mathbf{w}||}\right) \\ \hline \partial \mathbf{w} \end{bmatrix} - \begin{bmatrix} \frac{\partial(\frac{w_1}{||\mathbf{w}||} * (h_i * g_i))}{\partial \mathbf{w}} \\ \frac{\partial(\frac{w_2}{||\mathbf{w}||} * (h_i * g_i))}{\partial \mathbf{w}} \\ \vdots \\ \frac{\partial(\frac{w_D}{||\mathbf{w}||} * (h_i * g_i))}{\partial \mathbf{w}} \end{bmatrix} \right] \\ for \quad 0 < y_i(\mathbf{w}.\mathbf{x_i^t} + b) \leqslant 1 \\ I + \frac{1}{2\sqrt{2\pi}\sigma_{ck}} \sum_{i \in n_{sv}} \left[\begin{bmatrix} x_i^1 \partial\left(\frac{f_i}{||\mathbf{w}||}\right) \\ \hline \partial \mathbf{w} \\ x_i^2 \partial\left(\frac{f_i}{||\mathbf{w}||}\right) \\ \hline \partial \mathbf{w} \\ \vdots \\ x_i^D \partial\left(\frac{f_i}{||\mathbf{w}||}\right) \\ \hline \partial \mathbf{w} \end{bmatrix} - \begin{bmatrix} \frac{\partial(\frac{w_1}{||\mathbf{w}||} .* (h_i * g_i))}{\partial \mathbf{w}} \\ \frac{\partial(\frac{w_2}{||\mathbf{w}||} * (h_i * g_i))}{\partial \mathbf{w}} \\ \vdots \\ \frac{\partial(\frac{w_D}{||\mathbf{w}||} * (h_i * g_i))}{\partial \mathbf{w}} \end{bmatrix} \right] \\ for \quad y(\mathbf{w}.\mathbf{x_i^t} + b) < 0 \end{cases}$$

and the gradient is

$$\nabla = \begin{cases} 2 * \mathbf{w^t} - \frac{1}{\sqrt{2\pi}\sigma_{ck}} \sum_{i \in n_{sv}} f_i * \frac{\mathbf{x_i^t}}{||\mathbf{w}||} - \frac{f_i}{||\mathbf{w}||} * g_i * \frac{\mathbf{w^t}}{||\mathbf{w}||} \\ for \quad 0 < y_i(\mathbf{w}.\mathbf{x}_i + b) \leqslant 1 \\ 2 * \mathbf{w}^t + \frac{1}{\sqrt{2\pi}\sigma_{ck}} \sum_{i \in n_{sv}} f_i * \frac{\mathbf{x_i^t}}{||\mathbf{w}||} - \frac{f_i}{||\mathbf{w}||} * g_i * \frac{\mathbf{w}^t}{||\mathbf{w}||} \\ for \quad y_i(\mathbf{w}.\mathbf{x}_i + b) < 0 \end{cases}$$

where

$$\begin{cases} h_i & = \frac{f_i}{||\mathbf{w}||} \\ \frac{\partial h_i}{\partial w} & = \frac{f_i'}{||\mathbf{w}||} - \frac{\mathbf{w^t}}{||\mathbf{w}||^3} f_i \\ f_i & = exp\left(- \left(\frac{|\mathbf{w}.\mathbf{x}_i^t + b|}{||\mathbf{w}||\sqrt{2}\sigma_{ck}} \right)^2 \right) \\ g_i & = \frac{\mathbf{w}.\mathbf{x_i^t} + b}{||\mathbf{w}||} \end{cases}$$

The above completes the mathematical formulation for a simple newton based method to be applied to the model. The next section focuses on the implementation difficulties that were faced while implementing the above in MATLAB.

Implementation details

Certain implementation details about the computational model of the above solution are worth noting. The specific problem that was faced was that of the scaling of the parameter w. The conditions in equation number 1.1 depend on the value of the expression $y(\mathbf{w}.\mathbf{x}_i^t + b)$. This expression needs to be normalized by $||w||$ to effectively apply this condition. This is a concept similar to the one used in designing the maximum margin definition for the SVM.

Secondly the second order derivative shown to be analytically existent was not Positive Semi Definite (PSD). Damping it with a regularizer is not a good solution as heavy damping is required to make it PSD and we effectively lose the second order information. A quasi newton method (L-BFGS) method is used instead, and is documented here[15]. It was effectively applied and achieved results good enough to be mentioned in this work. Also the bias term is simply solved for by extending the dimensionality of the data matrix X and w by one and letting the last term of w be the bias[5]. The implementation was done in MATLAB.

Chapter 3

ANALYSIS AND APPLICATIONS

There are three experiments that were carried out in this thesis. First with a synthetic dataset then with a Glioblastoma Multiform dataset and finally a lung cancer dataset. Cross-validation techniques are used for real life datasets in the last two cases. P-values, ROC/AUC analysis and confidence intervals are measured where needed.

Analysis using synthetic data sets

This experiment was done on Gaussian synthetic data sets. The RSVM does in no way take advantage of the fact that the data is Gaussian. A fact also demonstrated by to its performance on real life data sets in later datasets whose distribution is unknown. Gaussian noise of a predefined variance was added to the original distribution and the variance was increased slowly to test the effects of noise and outliers on the classifiers. Since the error of the classifiers is measured against the distribution we generate the data such that it has a diagonal and equal covariance to measure the error analytically[5]. It should be noted that doing the experiment with synthetic data helps in evaluating the classifier because we know the lower bound of the error and it is important to avoid cross validation which is not a unbiased estimator for the variance of the error for small sample datasets [2].

For two equal sized circular Gaussian the error of a classifier against a distribution can be calculated analytically as:

$$\int_{\boldsymbol{d}_{\mu 1}}^{\infty} exp(-x^2)dx + \int_{\boldsymbol{d}_{\mu 2}}^{\infty} exp(-x^2)dx$$

where $\boldsymbol{d}_{\mu 1}$ and $\boldsymbol{d}_{\mu 2}$ is defined as the euclidean distance of the classifier's hyperplane from the mean of the 2 classes of the data distributions. A Kolmogorov-Smirnov test was performed on the error samples of sufficient size to determine statistical

Variable name	Description	Symbol	Variable values for experiment
Sigma_data	This parameter is used to generate Gaussian data (circular).	s	[0.5, 0.6, 0.7, 0.8, 0.9, 1]
Sigma_noise	This is the noise that is present in the data per class	s_n	[0.5, 0.2, 0.1, 0.05];
Sample_size	Number of data points per class	N	[10,20, 50]
number of iterations	number of samplings of data set to run the classifier on each time	K	250
Dimensions	Dimensionality of the data	d	2,20

Table 3.1: The table showing the parameters for the experiment carried out on the synthetic datasets.

significance of the errors. For a given data set the training data is sampled K times and the errors against the distribution are collected.

The classifiers used in this experiment are the sigma classifier, the linear SVM and the RSVM. The pseudo code for the experiment is shown in Algorithm 3.1. Table 3.1 shows the parameters used for the experiment.

Experimental Procedure

This section explains how the experiments were exactly done as per listed in the pseudo code in Algorithm 3.1. The main loop begins on line 9. The hyper parameters for the R-SVM and Linear SVM are chosen by looking at the best performing classifier against the true distribution and not by any Cross validation technique. Since the classifiers are a pure optimization problem where in the functions are convex the machines are guaranteed to converge. The errors of these classifiers are measured against the true distribution (μ, s). All the errors in the

are collected and written to a disk to be further analyzed. There are $N * s * s_n$ files produced, each containing K errors.

The errors are then used to obtain a confidence plot shown in figure (3.1). It should be noted that the sigma classifier does not require hyperparameter term, all its parameters are derived from the data itself[11], also since it is a variant of an LDA based classifier there is no regularizer term. The results obtained are shown in figure (3.1). The experiment is in conformity of the main objective of this thesis which says that irrespective of the fact weather outliers are present or not, the error against the true data distribution must be minimal for a good classifier. Thus even though we add outliers by adding the s_n component to the Gaussian distribution, the error of a good classifier against the true distributions should be minimal and robust to these outliers.

Results

Figures (3.1) demonstrate the results for the previous section. As you can see the RSVM in does better than the Sigma classifier and both do better than the Linear SVM. The errors disappear as the data samples increase and the variance decreases. The superior performance of the Gaussian Error Functions chosen to represent the loss function demonstrates that there is room for improvement as far as outlier robustness in classifiers is concerned.

In figure (3.1) below there are three sub figures; Each for a fixed sample size N. In each sub figure, the X-axis is the increasing variance of the data sets. As the variance increases the error rate, represented on the Y-axis goes up. The "staircase" like steps in the figure are formed because for each variance value $i \in s$ we add a noise component $j \in s_n$. The p-values with some analysis are in the appendix section and are statistically significant.

Algorithm 3.1 Pseudo-code for the experiment carried out on the synthetic dataset

```
K = 250 % Samples to collect for significance test
s = [0.5, 0.6, 0.7, 0.8, 0.9, 1];
s_n = [0.5, 0.2, 0.1, 0.05];
N = [10,20,50,100,200,500];

for n = N
 for i = s
  for j = s_n
        errors = array(size(K),4);
        for k = 1:K
              X,Y = generateData(i,j, mean_data);
              model_svm   = LinearSVM(X,Y);
              model_sigma   = SigmaClasifier(X,Y);
              model_rsvm = RSVM(X,Y);
              errors(k,:) = error_distribution(model_svm
                  ,model_rsvm, model_sigma );

        end
        h1 = kstest2(errors(:,1), errors(:,2), 0.05);
        h2 = kstest2(errors(:,2), errors(:,3), 0.05);
        save(new File(), 'errors', 'h1', 'h2');
  end
 end
end
```

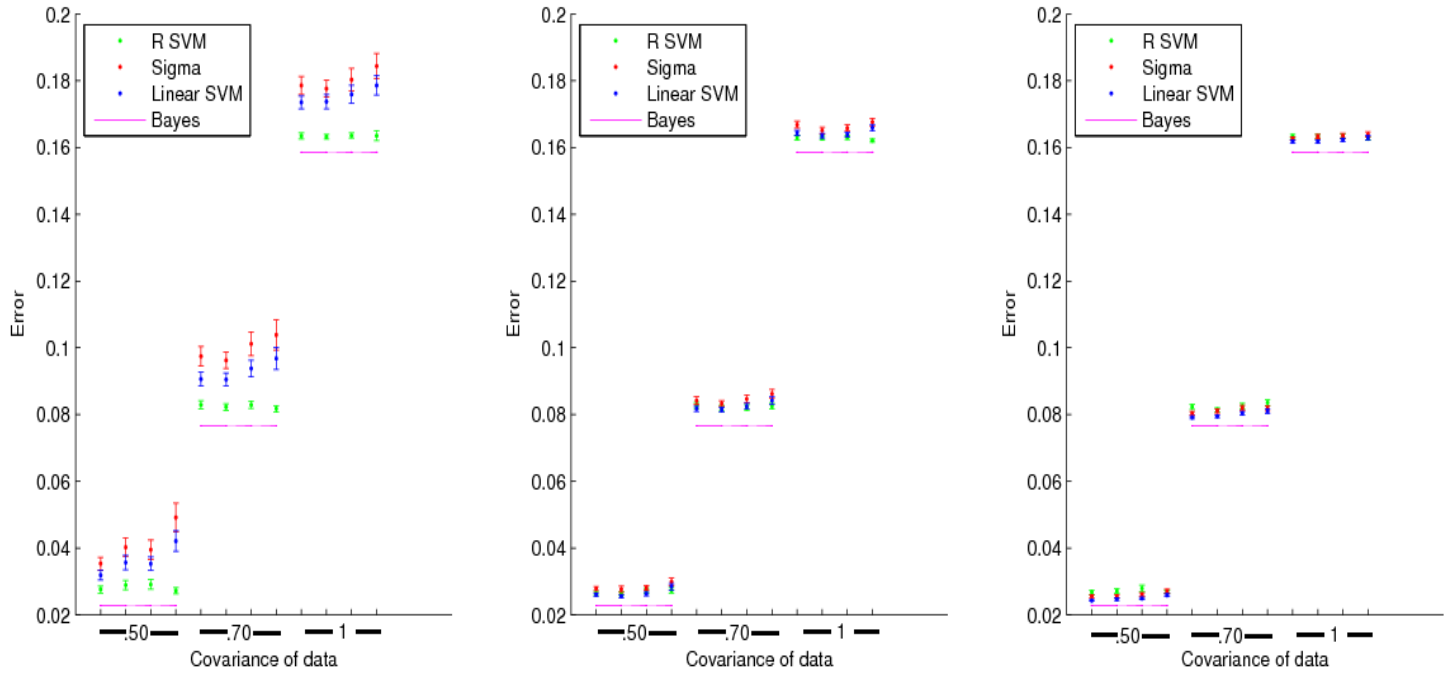

Figure 3.1: The result of the experiment on sythetic data showing superior RSVM performance and statistically significant error differences. The figure is for a 20-D Gaussian dataset. 250 samples were collected for estimating each confidence interval(96 percent). The three figures correspond to three different sample sizes of 10 , 30 and 50 samples respectively. In all figures the x-axis has twelve points, the points in a plot are divided into three groups, each group has four points. The variance of the three groups is 0.5, 0.7 and 1. Four noise variances are added to each group with values 0.05, 0.1, 0.2 and 0.5. As the noise increases so does the error within a group.

Applications

Glioblastoma Multiforme (GBM) data

The Glioblastoma multiforme (GBM) data set has 173 samples with 28 genes identified in the paper [23] as critical to distinguishing the classes apart. Glioblastoma multiforme is the most common and most aggressive type of primary brain tumor in humans, involving glial cells and accounting for 52% of all parenchymal brain tumor cases and 20% of all intracranial tumors. They are the most prevalent form of primary brain tumors according to a WHO study[14]. This experiment was performed by splitting the data at random into 2 parts with $\frac{4}{5}ths$ of the data or ~136 Sample for training and rest for testing. The classifiers were trained on these 136 samples with 10 fold cross validation and tested on the test data set. A non linear SVM with an RBF kernel was also tested for the sake of completeness of this thesis. There are 4 classes in the classifier so one vs. the rest performance is measured. This result was repeated 100 times for each of the datasets made as indicated in table 2 below. The Table 1 is graphically represented in Figure 3.2.

The results show the following observations

1. The Non linear SVM seems to over-fit the data for one of the datasets and is probably not very well performant due to over-fitting. This was observed due to the 0 training error but high test error.

2. The R-SVM and sigma classifiers perform almost similarly but slightly better than the existing SVM implementations.

Class division (4 classes)	Linear SVM	Sigma(±CM)	R SVM	NL SVM
Classical vs rest	0.087±0.006	0.046±0.008	0.078±0.008	0.20±0.018
Mychesmal vs rest	0.0923±0.007	0.037±0.018	0.052±0.009	0.30±0.011
Neural vs rest	0.083±0.0072	0.076± 0.014	0.164±0.0181	0.14±0.007
Pro Neural vs rest	0.077±0.0075	0.042±0.0064	0.049± 0.0094	0.27±0.025
Neural, Pro Neural vs rest	0.14±0.012	0.079±0.008	0.085±0.008	0.45 ±0.042

Table 3.2: The table shows the classification error for the 5 datasets. Thus row 1 represents the errors when the class Classical was taken in Class A and other 3 were taken in class B. The 5th row shows Neural and Pro Neural in one class and Classical and Mychesmal in the other. The sigma classifier has a lower mean error in all the cases. However, one should take caution in the fact that these are not statistically significant from other machines except in the case of row 1.

3. R-SVM and Sigma classifier agree that Neural and Pro Neural is well separated from Classical and Mychesmal.

4. It is also noted that R-SVM have an higher error than the sigma classifier on unbalanced classes as per the sample count ({'Neural': 26, Pro-Neural: 53, 'Classical': 38, 'Mesenchymal': 56}). Note that the neural (black) and classical (red) classes are more unbalanced than the others and are having higher error. This may indicate a susceptibility to class imbalance during training.

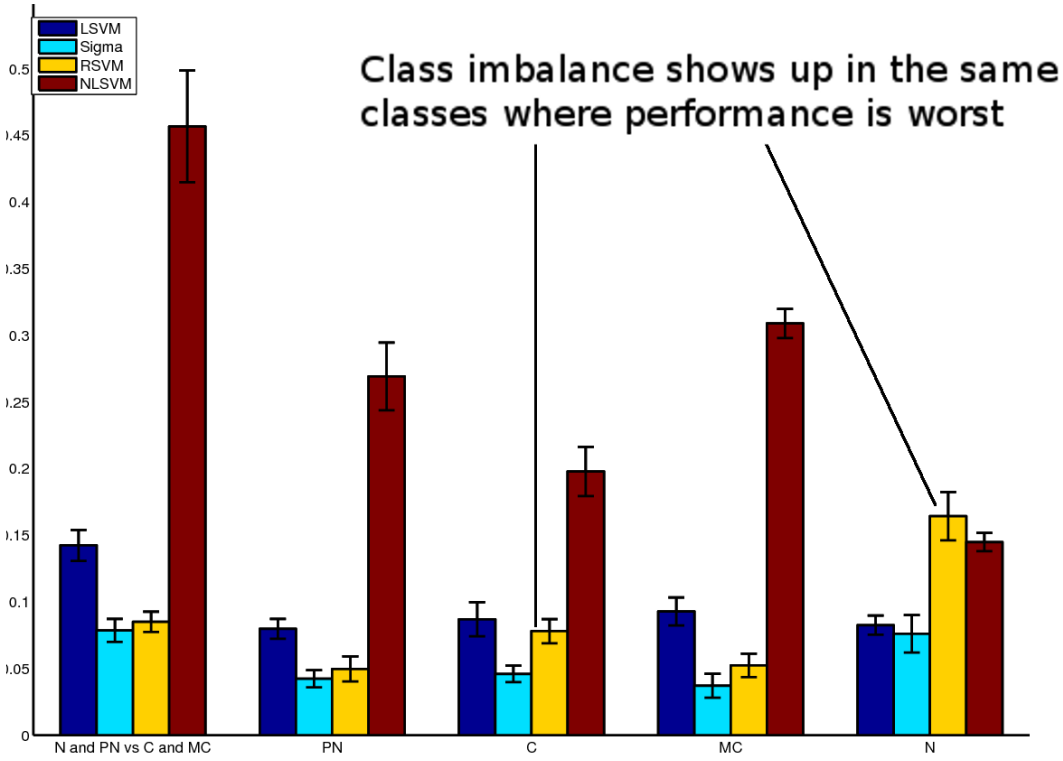

Figure 3.2: The figure shows the confidence intervals of the errors on the GBM data set. The X axis has four points for the four classifiers and the Y axis is the error or accuracy of the classifiers. See table 2 for more details. One can clearly see that the sigma classifier and the RSVM are almost indistinguishable when it comes to the confidence interval of the error.

Lung cancer data

A previous study on identifying Brain Metastaces (BM) possibility for non-small cell lung cancer (NSCLC) [1] by mi-RNA micro-array profiling was performed on samples from clinically matched NSCLC from patients with BM and without BM. Eight mi-RNAs were confirmed to be significantly differentially-expressed. Of these, expression of miR-328 and miR-330-3p were able to correctly classify BM+ vs. BM- patients.

The miR-328 and 330 were selected using the strong feature classifier. Left hand side of figure3.33.4 shows the prediction accuracy of the classifiers on the same data set, misclassifying BM+1 and BM-2. Then the same classifier is applied to the SHC Validation data and the result is shown on the right hand side of figure(3.3)(3.4) . Four samples were misclassified, SHC1+, SHC6- SHC15- and SHC14-, with 74% accuracy (specificity = 0.7300 and sensitivity = 0.75). All the samples from the training dataset were chosen for training the three SVM's in this case as only thirteen training examples were present. The error rates of the three classifiers are as indicated in table (3.3) below

Classifier	Linear SVM	Sigma classifier	RSVM
Error	0.33	0.2	0.26

Table 3.3: Results for the 3 SVM's on the lung cancer dataset. The sigma classifier and the RSVM performance is almost the same (differing only by 1 example which is misclassified)

The ROC and AUC figures for the trained RSVM are indicated in Figure (3.5). It is important to note that since all the training examples are used for the training and no cross validation is done, the ROC and AUC curves are used to measure of the classifier performance rather than the cross validation error used in the GBM dataset. The high area under the AUC curves typically indicates

the probability of the classifier to pick out positive examples correctly from the dataset. Although the use of ROC/AUC and its reliability as a metric for micro-RNA datasets is still a subject of debate[6]

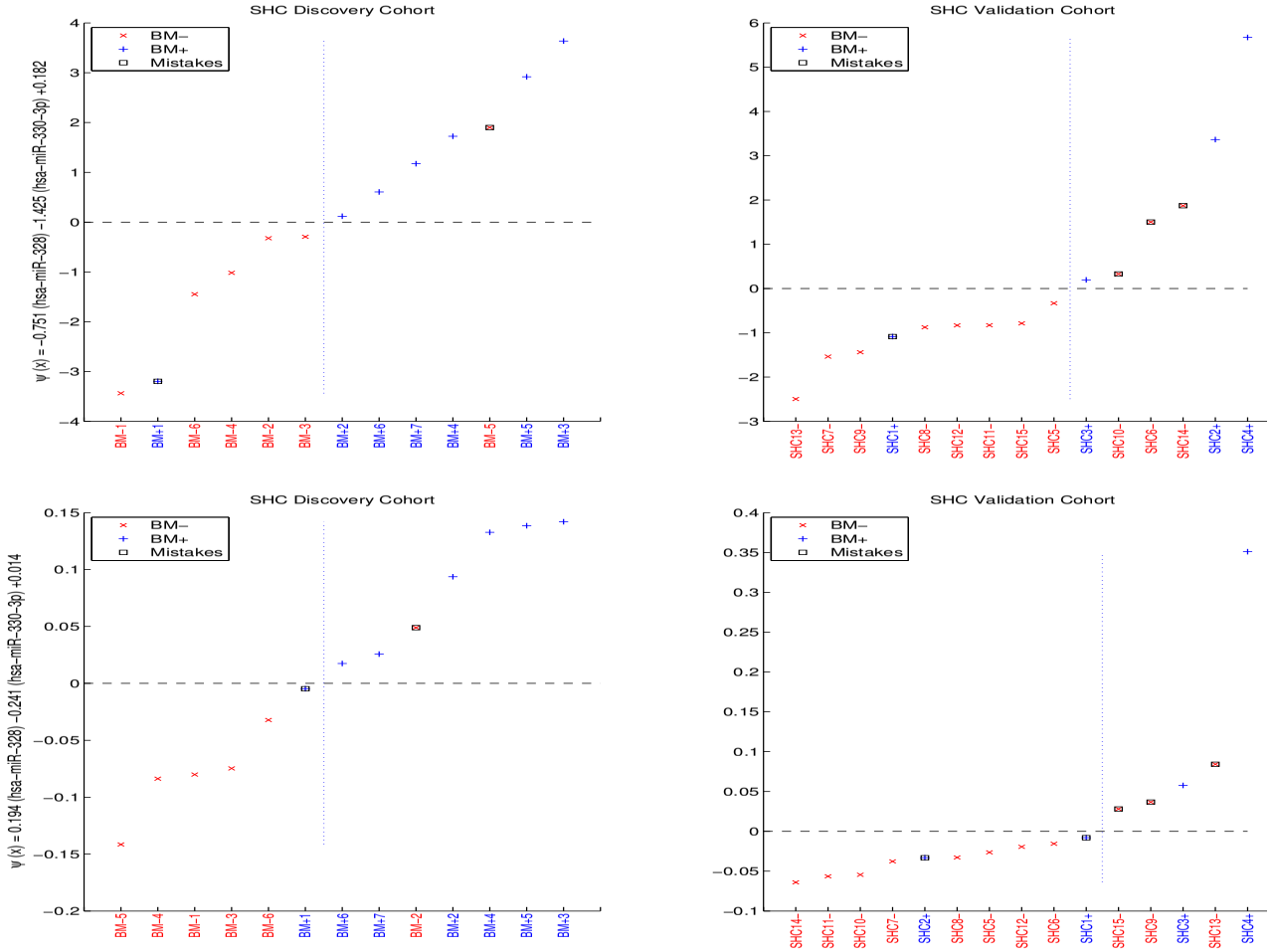

Figure 3.3: Prediction of the RSVM (top) and LSVM (bottom) on the lung cancer dataset as a function of the distance from the classifying hyperplane. Figure on the left shows the performance against the training dataset while those on the right are against the testing dataset for the two classifiers.

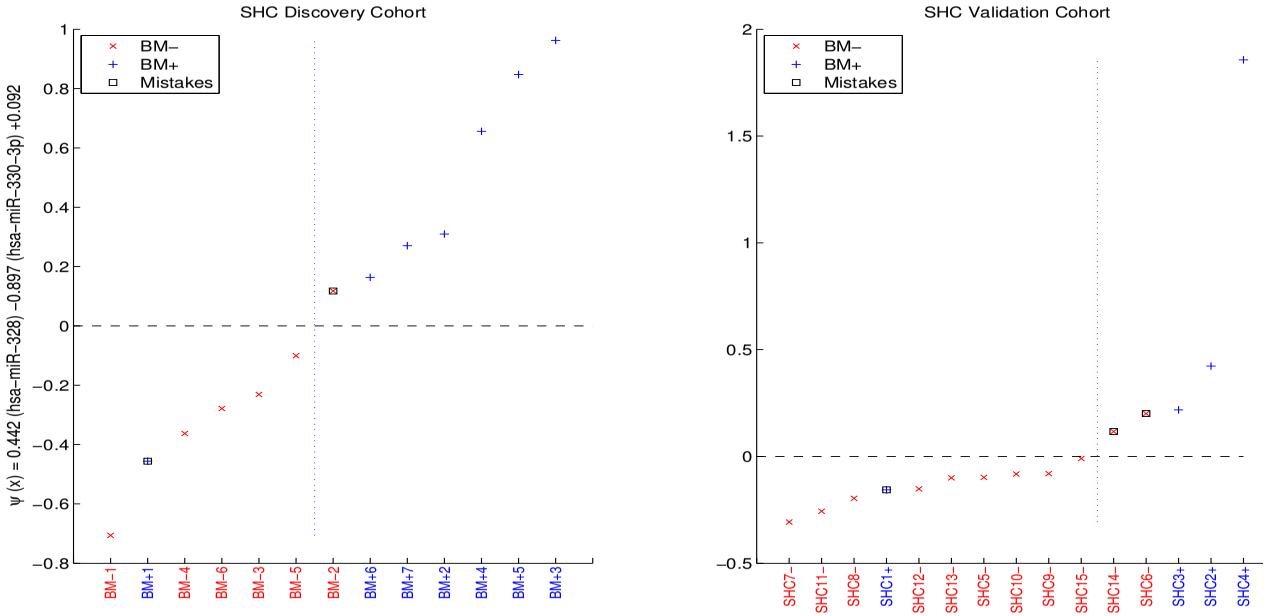

Figure 3.4: Prediction of the Sigma Classifier on the lung cancer dataset as a function of the distance from the classifying hyperplane. Figure on the left shows the performance against the training dataset while those on the right is against the testing dataset.

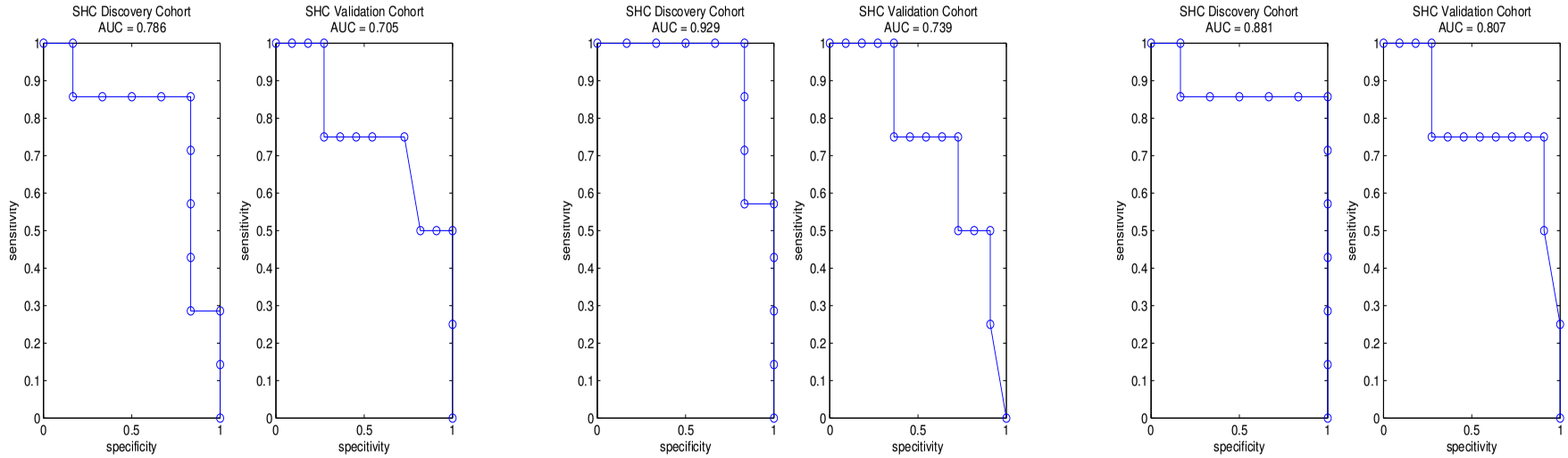

Figure 3.5: ROC and AUC measurements (Clockwise from top-left) for the RSVM, Linear SVM and Sigma Classifier on the Discovery and Validation cohorts carried on the lung cancer dataset.

Chapter 4

DISCUSSION AND FUTURE WORK

Even though the non linear SVM is demonstrated to overfit in some of the test data and previously cited work[17, 19], it is important in certain settings where over-fitting is not a problem. Thus the kernalization of the RSVM is a desirable feature. Also desirable is to use this loss function to select a strong feature set, something that was studied in previous works[11] but not here. Second order information needs to be implemented in the computational model and its lack of positive semidefinite nature needs to be analyzed, this can help in quadratic instead of super linear time needed for the machine to converge[15], but this is not a particularly pressing issue, given the results. Issues like class imbalance were experienced with GBM data set analysis that should also be addressed in the future.

This study helped in realizing that there is scope for improvement over the maximum margin classifiers. Some mathematical knowledge, specifically in the field of vector calculus was also gained. Important insights were obtained into the nature of second order optimization and quasi newton techniques and their space-time trade-offs. Also were studied a couple of important dimensionality reduction techniques like Information bottleneck[21] and Spectral clustering[16] that were not a part of this thesis but were used to analyze the data.

Biological data is unique in its nature because it is small sample and noisy, but getting the classification correct is all the more important, especially if clinical testing and cancer study depends on it. Thus the most important lesson learnt is, that one must analyze the data before selecting what algorithm should to be applied, every learning model has a weakness in some data setting and failure to take it into account can result in problems.

[1] S. Arora, A.R. Ranade, N.L. Tran, S. Nasser, S. Sridhar, R.L. Korn, J.T.D. Ross, H. Dhruv, K.M. Foss, Z. Sibenaller, et al. MicroRNA-328 is associated with non-small cell lung cancer (NSCLC) brain metastasis and mediates NSCLC migration. International Journal of Cancer.

[2] Y. Bengio and Y. Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. The Journal of Machine Learning Research, 5:1089–1105, 2004.

[3] O. Chapelle. Training a support vector machine in the primal. Neural Computation, 19(5):1155–1178, 2007.

[4] L. Devroye, L. Gyorfi, and G. Lugosi. A probabilistic theory of pattern recognition. Springer Verlag, 1996.

[5] R.O. Duda, P.E. Hart, and D.G. Stork. Pattern classification, volume 2. Citeseer, 2001.

[6] B. Hanczar, J. Hua, C. Sima, J. Weinstein, M. Bittner, and E.R. Dougherty. Small-sample precision of ROC-related estimates. Bioinformatics, 26(6):822, 2010.

[7] Y. Jia and C. Zhang. Local Regularized Least-Square Dimensionality Reduction. In Pattern Recognition, 2008. ICPR 2008. 19th International Conference on, pages 1–4. IEEE.

[8] M. Kaariainen. Generalization error bounds using unlabeled data. Learning Theory, pages 127–142, 2005.

[9] Thomas R.; Levinson David A. Kane. Differentiation of Vector Functions. Dynamics Online, Sunnyvale California, 1996.

[10] M. Kearns and D. Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. Neural Computation, 11(6):1427–1453, 1999.

[11] S. Kim, E.R. Dougherty, J. Barrera, Y. Chen, M.L. Bittner, and J.M. Trent. Strong feature sets from small samples. Journal of Computational Biology, 9(1):127–146, 2002.

[12] J. Langford. Tutorial on practical prediction theory for classification. Journal of Machine Learning Research, 6(1):273, 2006.

[13] John Langford. Clever methods of overfitting. "http://hunch.net/?p=22" and "http://hunch.net/?p=547".

[14] D.N. Louis, H. Ohgaki, O.D. Wiestler, W.K. Cavenee, P.C. Burger, A. Jouvet, B.W. Scheithauer, and P. Kleihues. The 2007 WHO classification of tumours of the central nervous system. Acta neuropathologica, 114(2):97–109, 2007.

[15] H. Matthies and G. Strang. The solution of nonlinear finite element equations. International Journal for Numerical Methods in Engineering, 14(11):1613–1626, 1979.

[16] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In Advances in Neural Information Processing Systems 14: Proceeding of the 2001 Conference, pages 849–856, 2001.

[17] N. Pochet, F. De Smet, J.A.K. Suykens, and B.L.R. De Moor. Systematic benchmarking of microarray data classification: assessing the role of nonlinearity and dimensionality reduction. Bioinformatics, 20(17):3185, 2004.

[18] C.E. Rasmussen. The infinite Gaussian mixture model. Advances in neural information processing systems, 12:554–560, 2000.

[19] S.J. Raudys and A.K. Jain. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13:252–264, 1991.

[20] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. Science, 290(5500):2323, 2000.

[21] N. Tishby, F.C. Pereira, and W. Bialek. The information bottleneck method. Arxiv preprint physics/0004057, 2000.

[22] V.N. Vapnik. The nature of statistical learning theory. Springer Verlag, 2000.

[23] R.G.W. Verhaak, K.A. Hoadley, E. Purdom, V. Wang, Y. Qi, M.D. Wilkerson, C.R. Miller, L. Ding, T. Golub, J.P. Mesirov, et al. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. Cancer Cell, 17(1):98–110, 2010.

[24] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3):37–52, 1987.

[25] L. Xu, K. Crammer, and D. Schuurmans. Robust support vector machine training via convex outlier ablation. In PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, volume 21, page 536. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

APPENDIX A
DERIVATIONS AND PROOFS

SIMILARITY OF THE PRIMAL AND DUAL FORMULATION.

Given a matrix $X \in R^{n*d}$ representing the coordinates of n points in d dimensions and a target vector $y \in R^n$, the primal RLS problem can be written as

$$\min_{w \in R^d} \lambda w^T w + ||Xw - y||^2 (\#1.1)$$

where λ is the regularization parameter. This objective function is popularly minimized for $w = (X^T X + \lambda I)^{-1} X^T y$ and its minimum is

$$y^T y - y^T X (X^T X + \lambda I)^{-1} X^T y. \ (\#1.2)$$

In typical SVM literature we introduce a slack variables $\xi = Xw - y$, the dual optimization problem then becomes

$$\max_{\alpha \in R^n} 2\alpha^T y - \frac{1}{\lambda} \alpha^T (XX^T + \lambda I)\alpha \ (\#1.3)$$

. The dual is maximized for ,$\alpha = \lambda (XX^T + \lambda I)^{-1} y$ and its maximum is

$$\lambda y^T (XX^T + \lambda I)^{-1} y \ (\#1.4)$$

The primal solution is then given by the KKT condition, $w = \frac{1}{\lambda} X^T \alpha$. The inverses of $XX^T + \lambda I$ and $X^T X + \lambda I$ and due to the Woodbury formula,

$$\lambda (XX^T + \lambda I)^{-1} = I - X(\lambda I + X^T X)^{-1} X^T$$

With this equality, we recover that primal (1.2) and dual (1.4) optimal values are the same, i.e. that the duality gap is zero.

APPENDIX B
TABLES

Table 1,2,3 and 4 depict the results that RSVM performs better than the LSVM and the sigma classifier for different sample sizes on the synthetic dataset. As sample size increase the other machines are able to generalize better and the performance gradually becomes similar as indicated by high p-values. The rows in each table are the standard deviation of the data for each class [0.5, 0.6, 0.7, 0.8, 0.9, 1] and the columns represent the noise variance [0.5, 0.2, 0.1, 0.05].

SYNTHETIC DATA RESULTS

Noise Variance↓	Data Variance→	0.5	0.6	0.7	0.8	0.9	1
0.5		2.3864e-05	9.8934e-11	6.3615e-08	2.2067e-28	7.6668e-10	5.588e-12
0.2		5.8127e-13	5.8127e-13	1.2487e-12	4.5327e-24	1.2487e-12	3.7589e-16
0.1		3.7589e-16	1.0319e-22	2.4631e-14	6.5658e-17	2.8653e-22	2.8653e-22
0.05		6.5658e-17	6.9303e-28	3.7589e-16	5.588e-12	7.8747e-22	1.537e-20

Table .1: Table showing clear p-value separation between the RSVM and Sigma classifier

Noise Variance↓	Data Variance→	0.5	0.6	0.7	0.8	0.9	1
0.5		0.0547	0.010442	0.062286	0.035785	0.080917	0.010404
0.2		0.052719	0.031245	0.69592	0.00035853	0.16675	0.080917
0.1		0.037666	0.044856	0.035785	0.035785	0.25622	0.37666
0.05		0.69592	0.25622	0.047454	0.00015139	0.019732	3.4897e-08

Table .2: Sample size increase to n = 30, The sigma classifier catches up and p-values are not statistically significant.

SYNTHETIC DATA RESULTS

Noise Variance↓	Data Variance→	0.5	0.6	0.7	0.8	0.9	1
0.5		2.6551e-12	1.0892e-14	5.588e-12	1.6114e-32	8.8564e-16	6.9961e-19
0.2		2.1421e-21	2.8653e-22	1.0585e-19	6.9961e-19	6.5658e-17	6.0843e-26
0.1		2.7022e-17	6.0843e-26	1.3443e-33	2.2067e-28	3.6786e-23	4.5327e-24
0.05		2.2067e-28	5.4938e-32	2.1543e-27	1.5668e-24	5.7675e-21	4.6782e-33

Table .3: Table showing separation between the LSVM and RSVM

Noise Variance↓	Data Variance→	0.5	0.6	0.7	0.8	0.9	1
0.5		0.003553	0.035785	0.026709	0.0023417	0.026709	0.062286
0.2		0.020781	0.010028	0.001675	0.0019732	0.0037224	1.8816e-06
0.1		0.0037666	0.010085	0.0053041	6.1353e-05	0.0023417	0.0037224
0.05		0.0037224	1.4656e-05	1.8816e-06	0.00054331	2.3864e-05	2.4e-11

Table .4: The p-values are still significantly high for LSVM vs RSVM at n = 30 samples