Linear and Non-Linear SVMs for Classification Problems

Daniel L. Marino
Department of Computer Science
Virginia Commonwealth University
Richmond, VA, USA
marinodl@vcu.edu

***Abstract*— In this paper we present the use of support vector machines (SVM) for classification problems. We demonstrate the use of different variations of SVM for classification of linearly and non-linearly separable data. First the Linear Maximal Margin Classifier for linearly separable data is presented; second, we show the extension for overlapping classes using a Linear Soft Margin Classifier; Third the non-linear SVM classifier is presented using Polynomial and Gaussian kernels. We show that Support Vector Machines are robust against outliers and how Cross-validation helps us to find SVM models that represent a good generalization of the dataset, preventing overfitting.**

Keywords- SVM, Kernel ,Crossvalidation

I. INTRODUCTION

Support Vector Machines are discriminant-based methods that instead of estimating the class probability density functions $P(C_i|\boldsymbol{x})$ to perform classification, SVM only makes an estimation of the class discriminants. Estimating the class densities is a harder problem than estimate the discriminant, therefore SVMs follow Vapnik's principle to never solve a more complex problem as a first step before the actual problem [1]

All the methods presented in this paper follow a similar quadratic program template whose objective is to maximize the *margin*, which is the distance from the separation boundary to the closest instances to it. Maximizing the margin allows us to get models that are a good generalization for the dataset.

In the case of a linearly separable dataset composed of two classes labeled as +1 and -1, the margin for a linear discriminant $y = w^T x + b$ can be expressed as $M = \frac{2}{\|w\|}$, [2]. Therefore, the problem of finding the optimal separating hyperplane that maximizes the margin can be expressed by the following program:

$$\begin{aligned} &\min \tfrac{1}{2}\|w\|^2 \\ &s.t\ \ d_i(w^T x_i + b) \geq 1\ ,\ \ i = 1,2,\dots,l \end{aligned} \tag{1}$$

Where (d_i, x_i) are the pairs of l training samples composed of input features $x \in \mathbb{R}^n$, and desired output $d_i \in \{-1,1\}$. $w^T x + b = 0$ is the separation boundary, with margin $M = \frac{2}{\|w\|}$, as said before. We can express this problem as an unconstrained optimization problem using Lagrange multipliers $\boldsymbol{\alpha} \in \mathbb{R}^l$:

$$L_p(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \tfrac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \boldsymbol{\alpha}^T(\boldsymbol{D} \circ (\boldsymbol{X}\,\boldsymbol{w} + b) - 1) \tag{2}$$

Where $\boldsymbol{X} \in \mathbb{R}^{l\,x\,n}$ is a matrix composed by all training samples arranged as row vectors $X = [x_1\ x_2\ \dots\ x_l]^T$. $D \in \mathbb{R}^l$ is a column vector composed by all desired ouputs d_i , $D = [d_1\ \ d_2 \dots d_l]^T$. The Hadamard product (element-wise product) is denoted with the $\circ$ symbol and we make a small abuse of notation by defining the summation of a matrix by a scalar ($\mathbf{C}$=$\boldsymbol{A} + b$) where the result $\mathbf{C}$ is a matrix of the same size of $\boldsymbol{A}$, where the scalar value is added to each element of $\boldsymbol{A}$, i.e. $C_{ij} = A_{ij} + b$.

Making use of the Karush-Kuhn-Tucker conditions, we can turn Eq. (2) into the following quadratic optimization problem that is equivalent to the program expressed in Eq. (2), but that is only expressed in terms of $\boldsymbol{\alpha}$, [2]:

$$\begin{aligned} &\min \tfrac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{H}\boldsymbol{\alpha} - \boldsymbol{p}^T\boldsymbol{\alpha} \\ &s.t.\begin{cases} \boldsymbol{D}^T\boldsymbol{\alpha} = 0 \\ \mathbf{0} \leq \boldsymbol{\alpha} \end{cases} \end{aligned} \tag{3}$$

Where $\boldsymbol{H} = (\boldsymbol{D}\boldsymbol{D}^T) \circ (\boldsymbol{X}\boldsymbol{X}^T)$, and $\boldsymbol{p} \in \mathbb{R}^l$ is a vector composed of l ones, $\boldsymbol{p} = [1\ 1\ \dots\ 1]^T$. Having found the $\boldsymbol{\alpha}$ that solves Eq. (3), we can express the optimal separation hyperplane $\boldsymbol{w}_o^T\boldsymbol{x} + b_o = 0$ as follows:

$$w_o = \sum_{i=1}^{l} \alpha_i d_i \boldsymbol{x}_i$$

As we can see, the discriminant is expressed as a weighted sum of the training samples, furthermore, some training samples $\boldsymbol{x_i}$ have a corresponding $\alpha_i = 0$, therefore, these samples do not contribute to the solution. The training samples $X_{SV} = \{\boldsymbol{x_i} | 0 < \alpha_i\}$ are called support vectors (i.e. those whose corresponding α_i is bigger than zero), so we can express our discriminant model using only a subset of the training dataset, this is a key point for non-linear classification. Thereby, the Linear Maximal Margin Classifier model can be expressed as follows:

$$w_o = \sum_{i=1}^{|X_{SV}|} \alpha_i d_i \boldsymbol{x}_i\ \ ;\ b_o = \frac{1}{|X_{SV}|}\sum_{i=1}^{|X_{SV}|} d_i - \boldsymbol{x}_i^T w_o \tag{4}$$

where $x_i \in X_{SV}$.

In the case of overlapping classes, we only have to make a small modification to the optimization problem in Eq. (3), we stablish an upper bound C for all alphas:

$$\min \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{H}\boldsymbol{\alpha} - \boldsymbol{p}^T\boldsymbol{\alpha} \quad s.t. \begin{cases} \boldsymbol{D}^T\boldsymbol{\alpha} = 0 \\ 0 \leq \boldsymbol{\alpha} \leq C \end{cases} \tag{5}$$

In this case, we make the following distinctions:

- $\boldsymbol{x_i}$ is a Support Vector if its corresponding α_i is bigger than zero, i.e. if $\boldsymbol{x_i} \in X_{SV} = \{\boldsymbol{x_i} \in X | 0 < \alpha_i\}$, we also define $D_{SV} = \{d_i \in D | \boldsymbol{x_i} \in X_{SV}\}$ and $\alpha_{SV} = \{\alpha_i \in \alpha | \boldsymbol{x_i} \in X_{SV}\}$. Every time that we refer to X_{SV}, D_{SV} or α_{SV} as matrices/vectors, we assume that the rows of the matrix are composed by the elements of the corresponding set.
- $\boldsymbol{x_i}$ is a Bounded Support Vector if its corresponding α_i is equal to C, i.e. if $x_i \in X_{BSV} = \{x_i \in X | \alpha_i = C\}$. Note that $X_{BSV} \subseteq X_{SV}$
- $\boldsymbol{x_i}$ is a Free Support Vector if its corresponding α_i is less than C, i.e. if $x_i \in X_{BSV} = \{x_i \in X | \alpha_i < C\}$. Note that $X_{FSV} \subseteq X_{SV}$

The optimal separation hyperplane in this case is defined as follows:

$$w_o = \sum_{i=1}^{|X_{SV}|} \alpha_i d_i \boldsymbol{x}_i \;\; ; \;\; for\; \boldsymbol{x}_i \epsilon X_{SV} \tag{6}$$

$$b_o = \frac{1}{|X_{FSV}|} \sum_{i=1}^{|X_{FSV}|} d_i - \boldsymbol{x}_i^T w_o \;\; ; \;\; for\; \boldsymbol{x}_i \epsilon X_{FSV} \tag{7}$$

This model is called Linear Soft Margin Classifier. To extend this model to the non-linear case, we notice that we can express the output y of the model to predict the class of a new sample $\boldsymbol{x}'$ as follows:

$$y = w_o^T x' + b_o$$

$$y = \sum_{i=1}^{|X_{SV}|} \alpha_i d_i \boldsymbol{x'}^T \boldsymbol{x_i} + b_o \;\; ; \;\; \{\alpha_i, d_i, x_i | \boldsymbol{x}_i \epsilon X_{SV}\}$$

where the $\boldsymbol{x}_i \epsilon X_{SV}$ are the support vectors found in the training phase and α_i and d_i are its corresponding α and desired value. If data is non-linearly separable, we can use non-linear basis functions $\boldsymbol{\phi}(\boldsymbol{x})$ to map the input space to a higher order space where luckily the data will be linearly separable, therefore we have:

$$y = \sum_{i=1}^{|X_{SV}|} \alpha_i d_i \boldsymbol{\phi}(\boldsymbol{x'})^T \boldsymbol{\phi}(\boldsymbol{x_i}) + b_o$$

This is called the *dual representation*, and allows us to express the model using Kernels:

$$y = \sum_{i=1}^{|X_{SV}|} \alpha_i d_i K(\boldsymbol{x'}, \boldsymbol{x_i}) + b_o = \boldsymbol{K}(\boldsymbol{x'}, X_{SV})(\alpha_{SV} \circ D_{sv}) + b_o \tag{8}$$

The kernel $K(\boldsymbol{x'}, \boldsymbol{x_i}) = \boldsymbol{\phi}(\boldsymbol{x'})^T \boldsymbol{\phi}(\boldsymbol{x_i})$ allows us to map $\boldsymbol{x}$ to a new space where we can fit the linear model by using the nonlinear basis function $\boldsymbol{\phi}(\boldsymbol{x})$ without the trouble of actually computing $\boldsymbol{\phi}(\boldsymbol{x})$. In this paper we make use of Polynomial kernels and Gaussian kernels that are defined as follows:

- Polynomial kernel of degree d:
$$K(\boldsymbol{x'}, \boldsymbol{x_i}) = \left[\boldsymbol{x'}^T \boldsymbol{x_i} + 1\right]^d$$
- Gaussian kernel:
$$K(\boldsymbol{x'}, \boldsymbol{x_i}) = \exp\left[-\frac{1}{2}(\boldsymbol{x'} - \boldsymbol{x_i})^T \Sigma^{-1} (\boldsymbol{x'} - \boldsymbol{x_i})\right]$$

To find the values of α_i on Eq. (8) we solve the same program expressed in Eq. (5), but in this case the Hessian matrix $\boldsymbol{H}$ defined as follows:

$$H_{ij} = d_i d_j K(\boldsymbol{x_i}, \boldsymbol{x_j}) \;\; ; \;\; \boldsymbol{x_i}, \boldsymbol{x_j} \epsilon X$$

$$\boldsymbol{H} = (\boldsymbol{D}\boldsymbol{D}^T) \circ \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) \tag{8}$$

where the matrix $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$ is sometimes referred as the Grammian matrix of the set of vectors $\{\boldsymbol{\phi}(\boldsymbol{x_i}) | \; \boldsymbol{x_i} \epsilon X\}$, and again, just to be clear, X is the set of training samples.

One important point to clarify is that when working with nonlinear kernels, we never calculate the optimal value for w_o, instead we store the Support Vectors and use the *dual representation* of the model (Eq. 8) to estimate the class of new data points.

II. Linear Maximal Margin Classifier

This section shows how a Linear Maximal Margin Classifier model can be used to find the optimal separation hyperplane (in the sense of maximum margin) for a linearly separable dataset. Figure 1 shows the dataset, the separation boundary along with its margin, and the support vectors found using Eq. (3),(4)

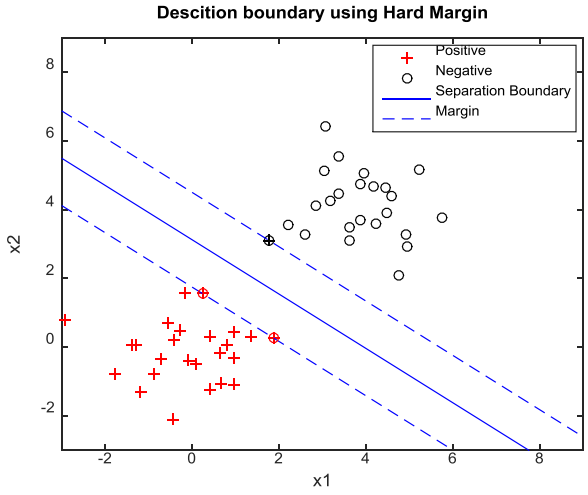

Figure 1: Separation boundary and margin found using a Linear Maximal Margin Classifier on a linearly separable dataset. The support vectors found are highlited using ⊕ symbols.

As can be seen, there are three support vectors that define the optimal separation boundary, the value of its alphas is shown in Table 1.

TABLE 1: ALPHA VALUES FOR THE SUPPORT VECTORS FOUND ON THE LINEARLY SEPARABLE DATASET

$\boldsymbol{Support\ Vector}$	$\boldsymbol{\alpha_i}$
(1.8860, 0.2570)	0.0450
(1.7622, 3.1128)	0.4276
(0.2477, 1.5532)	0.3825

The optimal separation boundary that was found is defined by $w_o = [-0.574, -0.725]^T, b_0 = 2.268$, and the margin is $M = \frac{2}{\|w\|} = 2.163$.

The value of the decision function for the data points $[3, 4]$ and $[6, 6]$ is the following:

$$y = \begin{bmatrix} 3 & 4 \\ 6 & 6 \end{bmatrix} \boldsymbol{w}_o + b_o = \begin{bmatrix} -2.35 \\ -5.52 \end{bmatrix}$$

Which means that both points are classified as negative samples.

An interesting fact is that the hyperplane that defines the margin is defined by the equation $w_o^T x + b_o = \pm 1$. This is basically because of the constraint $d_i(w^T x_i + b) \geq 1$ that we imposed on the problem. One way to see how we get this result is by considering that the separation hyperplane is defined by $w_o^T x + b_o = 0$, and the margin is $M = 2/\|w_o\|$, therefore the hyperplane that defines the margin must be the solution of:

$$w_o^T \left[x \pm \left(\frac{w_o}{\|w_o\|} \frac{M}{2} \right) \right] + b_o = 0$$

where $\frac{w_o}{\|w_o\|}$ is the unit vector pointing in the direction of w_o. By replacing the value of M in the previous equation we get:

$$w_o^T \left[x \pm \left(\frac{w_o}{\|w_o\|} \frac{2}{2\|w_o\|} \right) \right] + b_o = \left[w_o^T x \pm \frac{w_o^T w_o}{\|w_o\|^2} \right] + b_o$$

$$[w_o^T x \pm 1] + b_o = 0$$

$$w_o^T x + b_o = \pm 1$$

III. LINEAR SOFT MARGIN CLASSIFIER

This section shows how a Linear Soft Margin Classifier model can be used to find the optimal separation hyperplane for a dataset with overlapping classes. In this section we used the same dataset than in the previous section, but we added the points $[3, 4]$ and $[6, 6]$ to the dataset. The label for $[3, 4]$ is +1 and the label for $[6, 6]$ is -1, therefore the sample $[3, 4]$ is the one that makes our dataset non-linearly separable.

Figure 2 shows the dataset, the separation boundary along with its margin, and the support vectors found using Eq. (5),(6) and $C = 0.1$. As can be seen, the discriminant is a good generalization for the dataset, even when the point $[3, 4]$ is being misclassified.

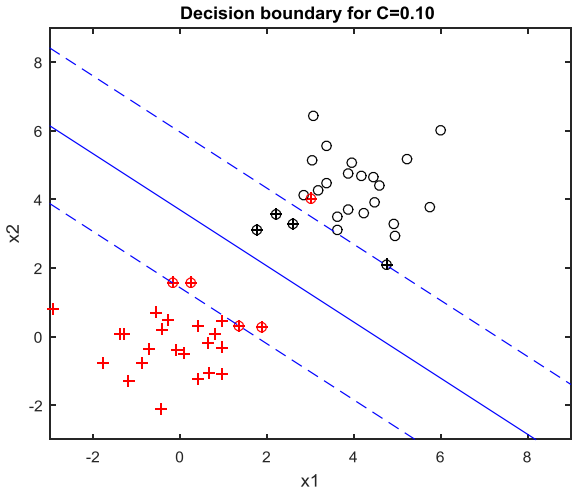

Figure 2: Separation boundary and margin found using a Linear Soft Margin Classifier on a dataset with overlaping classes with C=0.1.

All Support Vectors that are inside the margin, or wrongly classified have its $\alpha = C$. By visual inspection, in Figure 2, we can see that one SV is wrongly classified and 5 SV are inside the margin, these 6 SV constitute the set of Bounded SV, the remaining 3 SV constitute the set of Free SV.

In summary, the results that we got for C=0.1 are the following:

- $\mathrm{N_{sv}} = |X_{SV}| = 9$
- $\mathrm{N_{svfree}} = |X_{FSV}| = 3$
- $w_o = [-0.361, -0.441]$
- $b_0 = 1.628$
- Margin= 3.511
- $y = \begin{bmatrix} 3 & 4 \\ 6 & 6 \end{bmatrix} \boldsymbol{w}_o + b_o = \begin{bmatrix} -1.21 \\ -3.18 \end{bmatrix}$
- point $[3, 4]$ is being misclassified

Figure 3 shows the dataset, the separation boundary along with its margin, and the support vectors found using Eq. (5),(6) but this time with $C = 100000$. The results are the following:

- $\mathrm{N_{sv}} = |X_{SV}| = 5$
- $\mathrm{N_{svfree}} = |X_{FSV}| = 3$
- $w_o = [-0.368, -0.653]$
- $b_0 = 2.106$
- Margin= 2.667
- $y = \begin{bmatrix} 3 & 4 \\ 6 & 6 \end{bmatrix} \boldsymbol{w}_o + b_o = \begin{bmatrix} -1.611 \\ -4.023 \end{bmatrix}$
- point $[3, 4]$ is being misclassified

As we can see, with $C = 100000$ we are also getting the point $[3, 4]$ misclassified, but the discriminant is still a good generalization for the dataset. The point $[3, 4]$ is likely to be an outlier, in fact, [6 6] could potentially be another outlier, but we can see that the discriminant that was found after adding this two

points to the dataset is actually very similar to the one found in section II (Figure 1), so we can see how the Linear Soft Margin Classifier is robust against outliers.

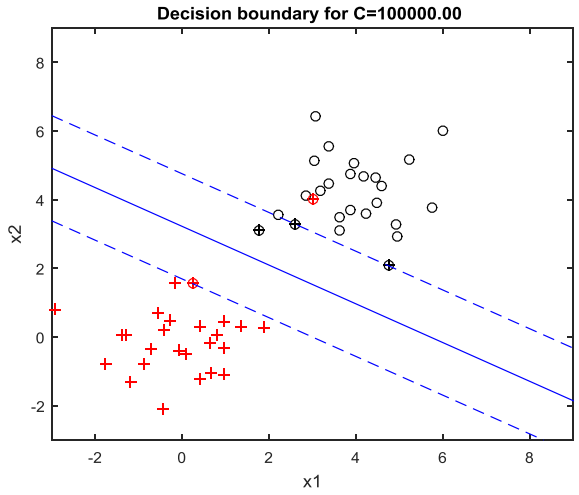

Figure 3: Separation boundary and margin found using a Linear Soft Margin Classifier on a dataset with overlaping classes with C=100000

IV. NON-LINEAR SVM MODEL

This section shows how a non-linear SVM model together with cross-validation can be used to find a discriminant that generalizes well for a non-linearly separable dataset with overlapping classes.

We first show the results that we got using a Polynomial Kernel. Figure 4 shows the dataset, the separation boundary and the support vectors found using Eq. (5),(8),(9). The separation boundary being shown corresponds to the best set of hyper-parameters (C, d) found using 5-fold cross-validation. For training, the Grammian $\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X})$ was calculated as $\boldsymbol{K}(\boldsymbol{X},\boldsymbol{X}) = (\boldsymbol{X}\boldsymbol{X}^T + 1)^{\boldsymbol{d}}$, where the exponentiation by d is done element-wise.

The results using polynomial kernel are:

- Minimum error found using 5-CV: 13%
- Average number of SV for best discriminant found: 30.4
- Best C= 10
- Best d=2

Figure 5 shows the decision surface of the model. We can see that although we got an average of 13% of the test samples misclassified, the discriminant is a good generalization for the dataset. Figure 6 shows the error surface that we got for different values of C and d, we can see how the error increases when the degree of the polynomial is big, this is basically because very high polynomials tend to produce overfitting.

Figure 7 shows the dataset, the separation boundary and the support vectors found using Eq. (5),(8),(9) and a Gaussian kernel. For the Gaussian kernel, we used a diagonal matrix as covariance matrix with the same standard deviation σ in all dimensions:

$$K(\boldsymbol{x}', \boldsymbol{x_i}) = \exp\left[-\frac{1}{2}(\boldsymbol{x}' - \boldsymbol{x_i})^T \frac{I}{\sigma^2}(\boldsymbol{x}' - \boldsymbol{x_i})\right] = \exp\left[-\frac{1}{2\sigma^2}\|\boldsymbol{x}' - \boldsymbol{x_i}\|^2\right]$$

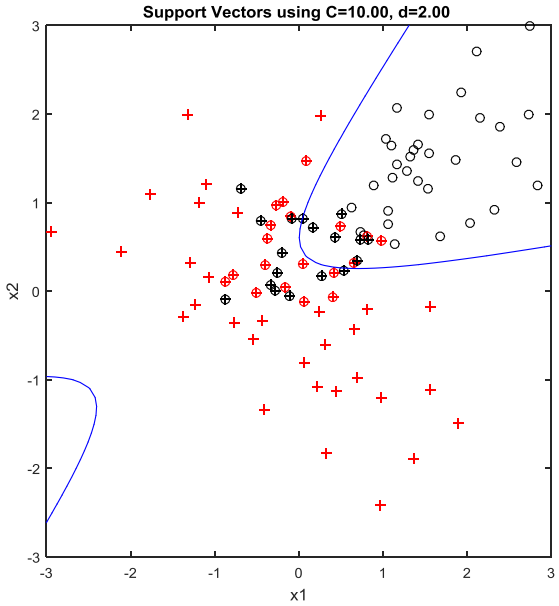

Figure 4: Best Separation boundary found by using 5-fold Cross-validation and polynomial kernel

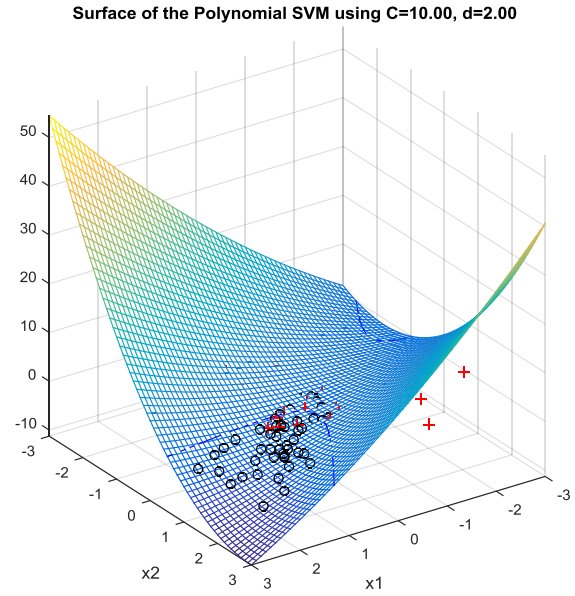

Figure 5: Best desition surface found using cross validation for polynomial kernel

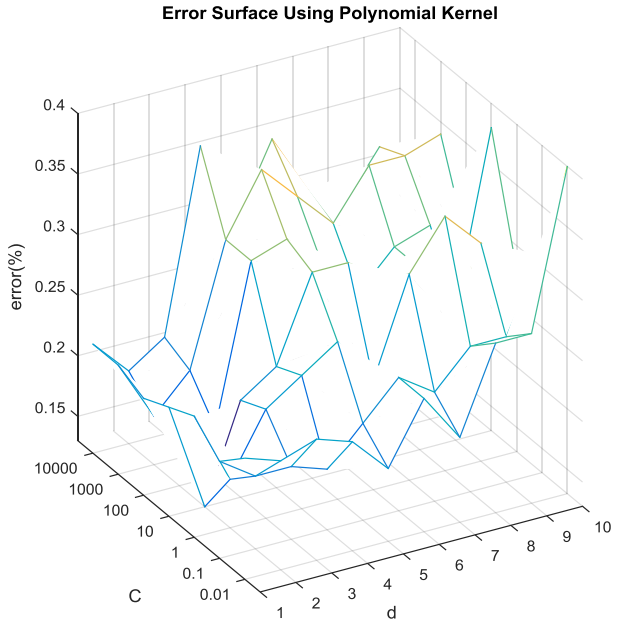

Figure 6: Error surface for different values of C and d, using a Polynomial Kernel

For calculating the Grammian $\boldsymbol{K(X,X)}$, we used the following trick for calculating the squared norm of $\boldsymbol{x_i - x_j}$:

$$\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2 = (\boldsymbol{x_i} - \boldsymbol{x_j})^T(\boldsymbol{x_i} - \boldsymbol{x_j}) = \boldsymbol{x_i^T x_i} - 2\boldsymbol{x_i^T x_j} + \boldsymbol{x_j^T x_j}$$

Calculating $\|\boldsymbol{x_i} - \boldsymbol{x_j}\|^2$ as $\boldsymbol{x_i^T x_i} - 2\boldsymbol{x_i^T x_j} + \boldsymbol{x_j^T x_j}$ actually requires more floating point operations, but allows us to write the calculation of the Grammian as a vectorized operation (with singleton expansion enabled), which makes its implementation in Matlab a little bit faster and cleaner.

The results using Gaussian kernel are:

- Minimum error found using 5-CV: 16%
- Average number of SV for best discriminant found: 35.4
- Best C= 1000
- Best sigma= 5

Figure 8 shows the decision surface corresponding to the best discriminant found, Figure 9 shows the Error surface that we got with 5-Fold Cross-validation. It is interesting to see that the decision surface and decision boundary found using Polynomial kernel and Gaussian kernel are actually very similar, which suggest that the models are a good generalization (low variance).

It is important to highlight that we actually got 3 different pairs of (C, sigma) with the same minimum error reported by Cross-validation (16%). Table 2 shows these tree values together with their corresponding average number of support vectors. We selected the pair C=1000 and sigma=5 following Occam's razor principle: "Among competing hypotheses, the one with the fewest assumptions should be selected", in this case we selected this pair because is the one with fewer average number of support vectors and bigger sigma, which translates into the "simpler" model between the three.

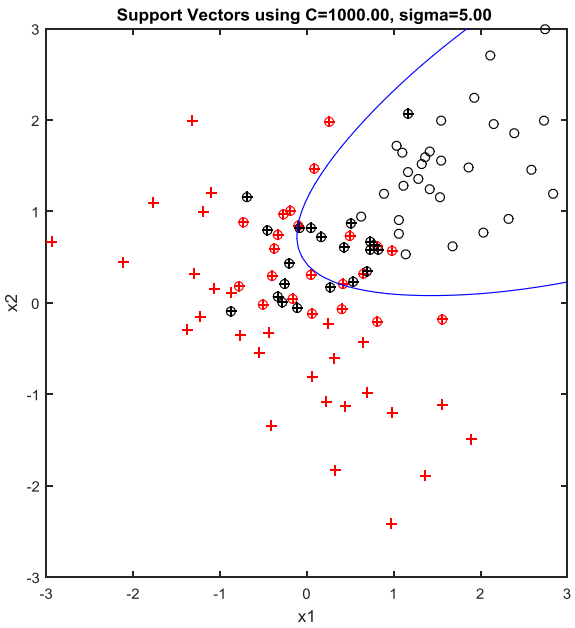

Figure 7: Best Separation boundary found by using 5-fold Cross-validation and Gaussian kernel

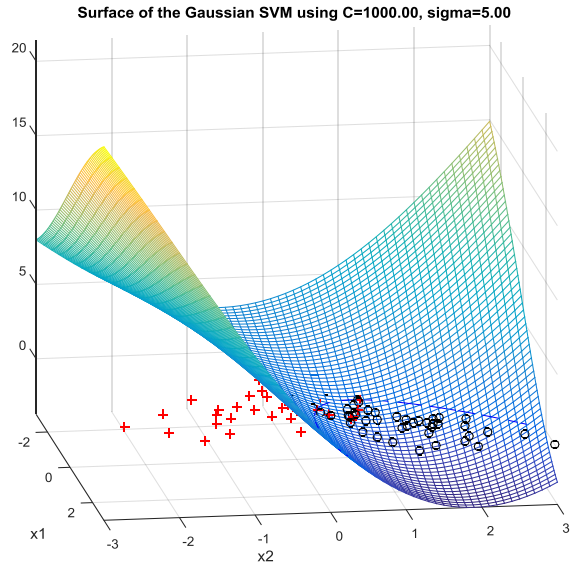

Figure 8: Best decision surface found using cross validation and Gaussian kernel

TABLE 2: C AND SIGMA VALUES WITH MINIMUM ERROR OF 16% REPORTED BY 5-FOLD CV.

C	Sigma	Average Nsv
1	0.75	50.2
10	1	38.4
1000	5	35.4

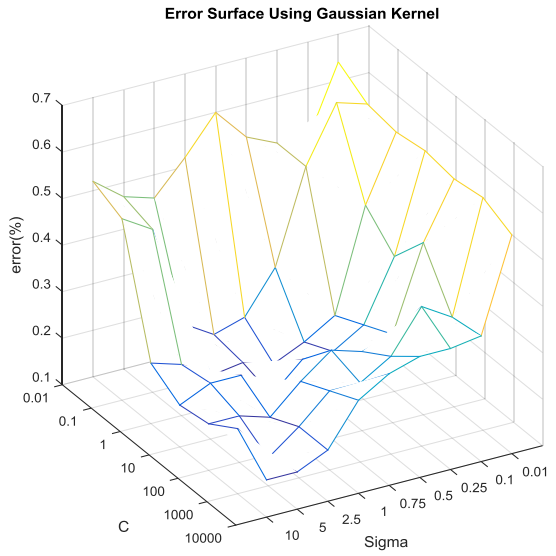

Figure 9: Error surface for different values of C and Sigma, using a Gaussian Kernel

Thanks to Cross-validation, we were able to find a discriminant model that generalizes well over the dataset. CV was the key to get good values for C, d and sigma, without CV is very easy to get an over-fitted model like the one shown on Figure 10 using a polynomial kernel of degree 10, which although its accuracy is of 96%, it is not a good generalization for the dataset.

Another example of an over-fitted model is shown in figure 11 and 12. These figures show a discriminant using Gaussian Kernels that has an accuracy of 100% on training data. We can see that the model basically selects almost all the points as support vectors, which allows it to have a perfect score on the training dataset, but it is not a good generalization for the problem and is likely to perform poorly on unseen data, a statement that we can support by looking to Figure 9, where this model produced a high error by testing it with Cross-validation.

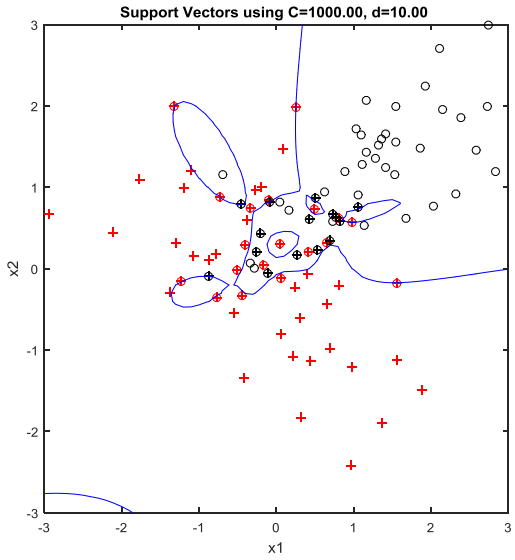

Figure 10: An example of an overfitted discriminant model that uses a polynomial kernel of degree 10 and has an acuracy of 96% on the training dataset

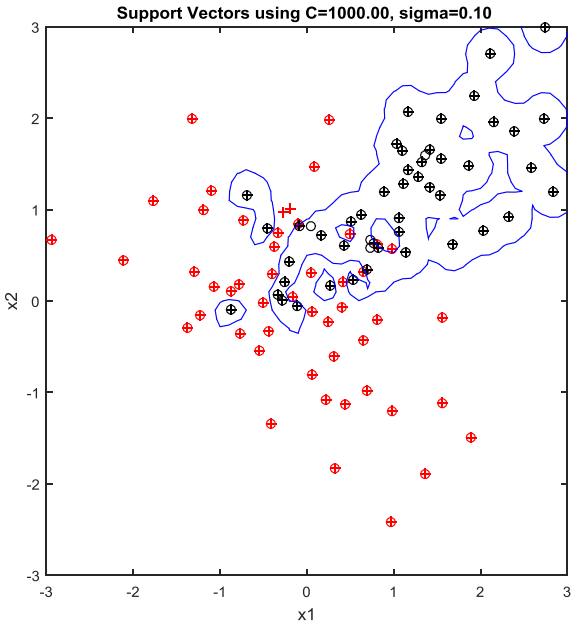

Figure 11: An example of an overfitted discriminant model that uses a Gaussian kernels and has an acuracy of 100% on the training dataset

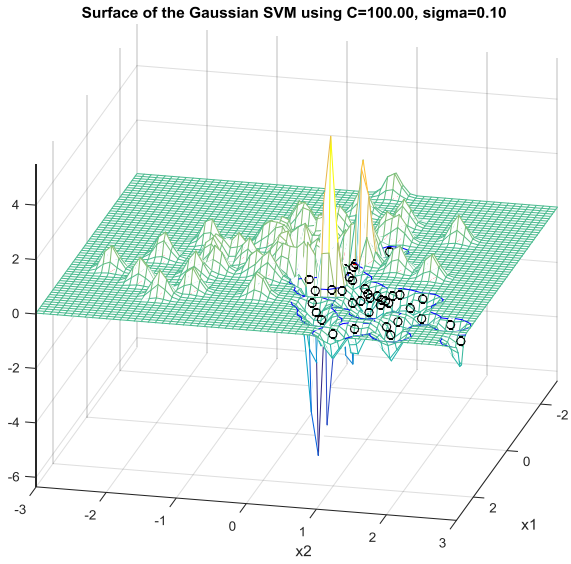

Figure 12: Decision surface of the overfitted discriminant model shown on figure 11.

V. Conclusions

We implemented and tested three SVM-based algorithms for solving classification problems with linearly and non-linearly separable data with overlapping. All three algorithms solve the problem by defining a quadratic program that aims to maximize the magin.

Our experiments showed that the algorithms are robust against outliers and are able to find good discriminant models when trained with help of Cross-validation.

When using non-linear SVM models, especially Gaussian Kernels, it is easy to being misled by a trained model that performs perfectly on training data but performs poorly on new data. Cross-validation proved to be essential to prevent overfitting when using polynomial or Gaussian kernels, it allowed us to find the models that are a good generalization for the dataset.

REFERENCES

[1] E. Alpaydin, "Introduction to Machine Learning", Second edition, MIT Press. 2010.

[2] T. Huand, V. Kecman, I. Kopriva, "Kernel Based Algorithms", Springer, pp. 11–60, 2005.