NATIONAL UNIVERSITY OF SINGAPORE

ME5404 Neural Networks

SVM Project
Support Vector Machine (SVM) for Classification of Spam Email Messages

---

Assignment – AY21/22 Semester 2

Name	Student Number	E-mail Address
Jin Hoontae	A0243155L	E0816449@u.nus.edu

EE5904/ME5404
Neural Networks
Project 1 AY2021/2022

Support Vector Machine (SVM) for Classification of Spam Email Messages

Jin Hoontae A0243155L

E0816449@u.nus.edu +65 8620 5429

Abstract

Multiple SVM kernels (linear, polynomial and radial basis function (RBF)) were utilised to examine their effectiveness of classifying the given dataset. The polynomial kernel oftentimes failed to satisfy *Mercer's* and *Karush-Kuhn-Tucker* (KTT) conditions thereby producing a poor accuracy for some values of the variables (p and C). On the other hand, it was observed that the linear kernel and the RBF kernel were able to classify the data with a high accuracy (≥90%). If the computation time is to be considered, the linear kernel could be the best option for the dataset given. However, if the stable high accuracy is desired, the RBF kernel is the best candidate under the assumption that the optimal hyperparameter values are used.

Support Vector Machine (SVM), *Mercer's* Conditions, *Karush-Kuhn-Tucker* (KTT) conditions, *Radial Basis Function* (RBF) kernel

1. Data Pre-processing

For Machine Learning (ML) algorithms, the data pre-processing is a necessary step as it makes disparate data sources uniform. There are generally two methods to realize the data uniformity; that is, *Data Normalization* and *Data Standardization*. The former is known as a scaling method that re-scales the values of input data to the range [0, 1]. This method considers minimum (X_{min}) and maximum (X_{max}) values of the feature, which can be expressed as:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad [1]$$

Although the latter is also a scaling method that re-scales the input values, it helps centre the values around the mean (u) with the standard deviation (σ). The technique can be denoted by **Eq. 2**:

$$X' = \frac{X - u}{\sigma} \quad [2]$$

In this project, the standardization was chosen over the normalization as it produced better accuracy during multiple trials. In the training data, the minimum and maximum feature values obtained were -0.9524 and 40.96, respectively after standardizing. As its previous minimum and maximum values were 0 and 9163, it was discovered that the standardization was successfully done. The same procedure was taken as well for the test and evaluation datasets.

2. Task 1

In Task 1, three different kernels were introduced. The goal was to compute the discriminant function for each kernel, if one exists, given the training dataset. For the hard margin, even though the value of C is assumed to be infinity in theory, 100 was used instead of 10^6. This is because it was found that the computation cost became extremely expensive. Nevertheless, the C value (=100) was adequately high for the hard margin to produce high accuracy. For the soft margin, the values (p and C) were given in advance.

2.1. Hard-Margin SVM with the linear kernel

The linear kernel ($K(x_1, x_2) = x_1^T x_2$) was computed with the standardized training dataset. For every SVM kernel, it is paramount to examine *Mercer's* condition; that is, the *Gram* matrix produced based on the kernel is a faulty candidate, if one of the eigenvalues is negative. However, an exception was made in this project that even if some of them were negative, if their absolute values were smaller than the pre-defined threshold value (=10^{-4}), they were assumed to be zero. This assumption had to be drawn as all the *Gram* matrices produced were found to be *not admissible*; that is, all of them contained at least one negative eigenvalue. Hence, by applying the new assumption, the *Gram* matrix for the linear kernel satisfied *Mercer's condition.* The hard margin with the linear kernel can separate two classes of data if and only if there exists a hyper plane that must meet the following condition:

$$w^T x + b = 0$$

where w is the weight vector, x is the feature data and b is the bias. In order to find an optimal hyper plane, we can make use of the dual problem and *Karush-Kuhn-Tucker* (KTT) conditions:

Given: $S = \{(x_i, d_i)\}$
Find: Lagrange multipliers $\{a_i\}$
Maximizing: $Q(a) = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j d_i d_j x_i^T x_j$ (for linear)
$Q(a) = \sum_{i=1}^{N} a_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j d_i d_j \varphi^T(x_i)\varphi(x_j)$ (for non-linear)
Subject to: (1) $\sum_{i=1}^{N} a_i d_i = 0$
(2) $a_i \geq 0$

The Lagrange multipliers could be obtained using the *MATLAB* built-in function (*quadprog*). After obtaining the multipliers, the existence of an optimal hyperplane was investigated.

For $\sum_{i=1}^{N} a_i d_i = 0$, the actual value obtained was $1.1418e^{-7}$. As the value was extremely close to zero, it was assumed that it satisfied the first condition. Moreover, the minimum and maximum values of the multipliers were 0 and 100, respectively. Thus, it could be said that the hard margin with the given linear kernel was capable of producing an optimal hyper plane with the given dataset as it met the two requirements ($\sum_{i=1}^{N} a_i d_i = 0$ and $a_i \geq 0$). Subsequently, weight (w_o) and bias (b_o) values were computed using the following equations:

$$w_o = \sum_{i=1}^{N} a_{o,i} d_i x_i \quad [3]$$

$$b_{o,i} = d_i - w_o^T x_i \quad [4]$$

$$b_o = \frac{\sum_{i=1}^{m} b_{o,i}}{m} \quad [5]$$

where o is the nth index that meets the condition ($0 < a_i < C$ *where* $C = 100$) and m is the total number of x_i with $0 < a_i < C$. 416 of 2000 in the Lagrange matrix was found to meet the condition, and a weight vector (57 × 1) and an average bias (-2.78) were obtained.

Table 1. Admissibility of the polynomial kernel with respect to p

p	2	3	4	5
Admissibility	O	O	X	X
$\sum_{i=1}^{N} a_i\, d_i$	$2.31e^{-6}$	$-3.70e^{-4}$	-0.0012	-0.0021
Min. a_i	0	0	0	0
Optimal Hyper Plane	O	O	X	X

Table 2. Weight and bias values with respect to p for the hard margin with the polynomial kernel

p	2	3	4	5
w_o	0.83	3.72	-3.09	-21.76
b_o	-1.77	-4.15	-2.28	-13.1

2.2. Hard-Margin SVM with the polynomial kernel

The same procedures were taken for the hard margin SVM with the polynomial kernel ($K(x_1, x_2) = (x_1^T x_2 + 1)^p$). **Table. 1** shows the admissibility of the kernel and the existence of optimal hyper planes with respect to different p values. As can be observed, the kernel begins to fail to satisfy *Mercer's condition* at p=4. And it was found that, when *Mercer condition* was not satisfied, the built-in *MATLAB* function (*quadprog*) returned *"The problem is Non-Convex*". This means that the problem consists of multiple minima, which complicates the process of finding the global minimum. If there is no minimum or a local minimum is chosen instead of global minimum in the function, it can be said that the optimization is likely to fail, which could also affect the accuracy for the classification. Hence, as for the existence of optimal hyper planes, the kernels with only p (=2 and 3) were able to satisfy the KKT conditions, expect for the ones with p (=4 and 5). This can be further explained by the values of $\sum_{i=1}^{N} a_i\, d_i$. As the polynomial value increases, it can be observed that its summation value tends to deviate from 0, which is not a desirable phenomenon. The values of weights and biases obtained are shown in **Table 2**. As can be seen in the table, the absolute values of weight and bias begin to increase rather significantly with the increasing value of p. One of the conditions required for an optimal hyper plane is that $||w||$ needs to be minimized as much as possible. Although we do not know whether the weight (=0.83) is minimized at p=2, when compared to other weight values, it can be presumed that the minimisation was done properly, which perhaps means that it could possibly produce fairly high accuracy compared to the rest

2.3. Soft-Margin SVM with the polynomial kernel

The admissibility of the soft margin with the polynomial kernel ($K(x_1, x_2) = tanh(x_1^T x_2 - 1)^p$) was examined. For the values (p and C) given in the assignment, all the kernels were found to fail to satisfy *Mercer's conditions*; that is, the *Gram* matrices contained at least one negative eigenvalue whose absolute value is bigger than the pre-defined threshold (=10^{-4}). Furthermore, it was discovered that all the non-linear kernels were non-convex, which made the optimization process of finding a global minimum for the Lagrange multipliers difficult. When C=0.1 or 0.6, the whole values in the vector (a_i) were greater than C, which implies that there was no single index that satisfied the condition ($0 < a_i < C$). This result gave rise to the erroneous computation subsequently; that is, the weight and bias could not be computed due to the absence of the desirable indexes of a_i. On the other hand, when C=1.1 or 2.1, the vector (a_i) contained multiple proper indexes. **Table 3** displays the values of weight and bias obtained when C=1.1 or 2.1 with the polynomial order (1 to 5). It was found that there existed no optimal hyper plane for all cases. This is because, firstly, the summation condition ($\sum_{i=1}^{N} a_i\, d_i = 0$) was not satisfied. Instead, their summation values were all significantly far away from zero.

Table 3. Weight, bias and summation values with respect to C and p values for the soft margin with the polynomial kernel

C	1.1				
p	1	2	3	4	5
w_o	$1.78e^{+3}$	$-1.88e^{+4}$	$2.72e^{+4}$	$-2.57e^{+3}$	$2.38e^{+3}$
b_o	-350	$1.31e^{4}$	-334	$1.98e^{3}$	-644
$\sum_{i=1}^{N} a_i\, d_i$	-841	$-2.13e^{+4}$	$-1.35e^{+4}$	$-3.46e^{+3}$	$-1.79e^{+3}$
C	2.1				
p	1	2	3	4	5
w_o	$2.78e^{+3}$	$-2.59e^{+3}$	$-4.20e^{+3}$	$-1.78e^{+3}$	$1.17e^{+4}$
b_o	-559	$2.21e^{+3}$	$-4.77e^{+3}$	$1.79e^{+3}$	$-5.28e^{+3}$
$\sum_{i=1}^{N} a_i\, d_i$	$-1.21e^{+3}$	$-3.10e^{+3}$	$-1.23e^{+4}$	$-2.77e^{+3}$	$-1.21e^{+4}$

** For C=0.1 and 0.6*, the variables (w_o and b_o) could not be obtained as there was no index that met the condition ($0 < a_i < C$) in the Lagrange multiplier vector.

Secondly, the values (w_o and b_o) were significantly high when, in fact, w_o needs to be minimized to find an optimal hyper plane. The proof can be found by looking at **Table 1&2**. As shown in the tables, a kernel obtains an optimal hyper plane when its value is low.

Thus, a conclusion could be drawn; that is, there would be occasions where the soft margin would produce a poor accuracy due to 1) the failure of satisfying the condition ($\sum_{i=1}^{N} a_i\, d_i = 0$), and 2) significantly high values of w_o. To investigate more in detail the existence of optimal hyper planes for each kernel, accuracy needed to be examined. Further analysis will be discussed in **Section 3** to examine the performance of all the kernels introduced thus far.

3. Task 2

In this section, the SVMs with the discriminant functions obtained in **Section 2** were implemented to classify the given training set and test set. **Table 4** summarizes the results of accuracy for the whole margins for the linear kernel and polynomial kernel with varying values of p and C. As can be observed in the table, the hard margin with the linear kernel obtained the highest accuracy (93.49%) for the test dataset, and its training accuracy (93.55%) is close as well, meaning that the SVM fits the data appropriately. On the other hand, even though the test accuracy obtained by the hard margin with the polynomial (p=2) kernel was 83.92%, its training accuracy was 99.5%. This implies that the model is overfitting the data. The cause of this overfitting behaviour might be because, at p=2, the kernel tends to memorize everything in the data without learning the pattern. Furthermore, as the polynomial value increases, its training and test accuracy begin to underfit the data to a great extent. Hence, for the given polynomial values of the hard margin, it obtains the most reasonable and best accuracy (85.25% and 83.92%) for the training set and test set at p=2, respectively. Nevertheless, the linear kernel still outperforms the polynomial kernel of the hard margin.

Table 4. Results of SVM classification

Type of SVM	Training accuracy				Test accuracy			
Hard margin with Linear kernel	93.55%				93.49%			
Hard margin with polynomial kernel	p=2	p=3	p=4	p=5	p=2	p=3	p=4	p=5
	99.5%	85.25%	75.05%	75.9%	83.92%	80.99%	71.94%	76.76%
Soft margin with polynomial kernel	C=0.1	C=0.6	C=1.1	C=2.1	C=0.1	C=0.6	C=1.1	C=2.1
p=1	N/A	N/A	87.7%	88.35%	N/A	N/A	89.32%	89.39%
p=2	N/A	N/A	39.1%	30.95%	N/A	N/A	34.57%	28.78%
p=3	N/A	N/A	80.8%	88.45%	N/A	N/A	84.18%	89.39%
p=4	N/A	N/A	33.65%	27.95%	N/A	N/A	30.79%	25.85%
p=5	N/A	N/A	88.35%	88.65%	N/A	N/A	89.39%	89.78%

** N/A* means that no index in the Lagrange vector (a_i) satisfied the condition ($0 < a_i < C$), resulting in the inability to compute the variables (w_o and b_o).

For the soft margin, no overfitting behaviour was observed. However, under-fitting was observed frequently when the polynomial value was even (=2 and 4), which signifies that it failed to find an optimal hyper plane. On the other hand, when its value was odd, high accuracy ($> 86\%$, generally) was obtained. The highest accuracy obtained for both training and test datasets was 88.65% and 89.78%, respectively at p=5. Hence, based on the accuracy results, the hard margin SVM with the linear kernel was found to be the most accurate and effective model among the other models for this specific dataset given. The main reason why it outperforms the polynomial kernels could be because the two classes (-1 and +1) in the dataset can be linearly separable, which does not require the polynomial-like complexity.

4. Task 3

4.1. Preparation for the evaluation dataset

An evaluation dataset was created by using the training set and test set. After combining the datasets into one, it was shuffled and 600 samples were extracted as advised in the guideline.

4.2. Radial Basis Function SVM construction

The *Radial Basis Function* (RBF) kernel was created to investigate its accuracy performance against the linear kernel for the evaluation dataset. The RBF kernel is known as the most generalised and widely used kernel due to its behaviour similar to the *Gaussian* distribution, which can be expressed by an equation:

$$K(x_i, x_j) = \exp\left(-\frac{\left||x_i - x_j|\right|^2}{2\sigma^2}\right) \quad [6]$$

where $\left||x_i - x_j|\right|$ is the Euclidean (L2-norm) distance between two points (x_i and x_j), and σ is the width variance (user-defined).

4.3. Mercer's conditions and existence of optimal hyper planes for the RBF kernel

Unlike the soft margin with the polynomial kernel, the RBF kernel was successfully able to satisfy *Mercer's condition* for all the values (C and σ). This is because the variance (σ) does not complicate the function graph regardless of its value as a higher value means merely the increase in the width of *Gaussian* distribution. Hence, their Lagrange multipliers could be obtained successfully as all the problems were convex consisting of a global minimum.

Table 4. Conditions of optimal hyper plans for the RBF kernel

C	0.1				
σ	1	2	3	4	5
min. a_i	0	0	0	0	0
$\sum_{i=1}^{N} a_i d_i$	$3.14e^{-4}$	$-6.38e^{-5}$	$1.60e^{-7}$	$1.23e^{-4}$	$2.16e^{-8}$
C	0.6				
σ	1	2	3	4	5
min. a_i	0	0	0	0	0
$\sum_{i=1}^{N} a_i d_i$	$-3.26e^{-5}$	$5.88e^{-7}$	$4.43e^{-5}$	$4.33e^{-8}$	$2.75e^{-5}$
C	1.1				
σ	1	2	3	4	5
min. a_i	0	0	0	0	0
$\sum_{i=1}^{N} a_i d_i$	$9.02e^{-7}$	$1.73e^{-6}$	$-2.62e^{-8}$	$4.69e^{-5}$	$7.01e^{-7}$
C	2.1				
σ	1	2	3	4	5
min. a_i	0	0	0	0	0
$\sum_{i=1}^{N} a_i d_i$	$4.76e^{-8}$	$-5.26e^{-6}$	$5.34e^{-9}$	$1.37e^{-5}$	-4.05^{-5}

Table 4 shows the summarisation of the KTT problem conditions: $0 \le a_i \le C$ and $\sum_{i=1}^{N} a_i d_i = 0$. As shown in the table, the minimum value of a_i for all the matrices was 0 and their maximum value was C, which realised the computation of weight and bias as no value was out of the range ($0 < a_i < C$). Moreover, all the summation values were extremely close to zero. From this observation, it can be said that the optimal hyper planes were found.

4.4. An accuracy comparison between the linear and RBF kernels

Accuracy comparison between the linear kernel and the RBF kernel was conducted to determine the winner. As can be seen in **Table 5**, no overfitting behaviour was observed for all the values with the RBF kernel except for the two cases (when C =1.1, $\sigma = 1$ and when C=2.1, $\sigma = 1$). Their training accuracy values were 99.5% and 99.8%, respectively. On the other hand, rather poor accuracy values (90.00% and 90.67%) were obtained for the evaluation set, creating an approximately 10% difference in the accuracy, which can be considered overfitting. The linear kernel was able to classify the evaluation dataset with an accuracy of 91.83%, and its training accuracy obtained was 93.55%. Despite its consistently high accuracy for the training, test and evaluation datasets, as it was discovered that the RBF kernel is a superior model in terms of the accuracy when proper values (C and σ), it is desirable to use the soft margin with the RBF kernel among the other kernels.

Table 5. An accuracy comparison between the linear kernel and the RBF kernel

Type of SVM	Training accuracy				Evaluation accuracy			
Hard margin with Linear kernel	93.55%				91.83%			
Soft margin with RBF kernel	C=0.1	C=0.6	C=1.1	C=2.1	C=0.1	C=0.6	C=1.1	C=2.1
σ=1	79.70%	98.8%	99.5%	99.8%	69.33%	85.33%	90.00%	90.67%
σ=2	92.15%	96.6%	97.80%	98.9%	89.83%	93.33%	95.17%	96.17%
σ=3	92.20%	94.9%	95.75%	97.5%	91.67%	93.00%	93.83%	95.00%
σ=4	91.80%	94.5%	94.90%	96.05%	90.67%	92.33%	93.33%	94.17%
σ=5	91.10%	93.6%	94.50%	95.15%	90.00%	92.00%	92.67%	93.67%