SVM for Classification of Spam Email Messages

AY 22/23 Semester 2 ME5404 Part II project I

Student name: Shen Xiaoting
Student number: A0263252L
Department: Mechanical Engineering
Student e-mail address: e1010610@u.nus.edu
Data: 8/4/2023

Contents

1 Abstract 3
2 Data pre-processing 3
3 Compute discriminant function 3
3.1 Admissibility of the kernels 3
3.2 A hard-margin SVM with the linear kernel 3
3.3 A hard-margin SVM with the polynomial kernel 4
3.4 A soft-margin SVM with the polynomial kernel 5
4 Training and testing accuracy calculation 6
4.1 A hard-margin SVM with the linear kernel 6
4.2 A hard-margin SVM with the polynomial kernel 6
4.3 A soft-margin SVM with the polynomial kernel 6
5 Radial Basis Function (RBF) kernel and implementation 7

1 Abstract

In this project, we implement the SVM to do spam data classification and understand the principles and issues of SVM for classification. We apply hard-margin SVM of linear kernel, hard-margin SVM of polynomial kernel and soft-margin of polynomial kernel. We calculated the discriminant function and get the optimal hyperplane of each case. Then we calculated the training and testing accuracy of each case with different hyperparameters and do comparison.

We also use the RBF SVM model to this dataset and form the evaluation set to assess the performance of RBF SVM model. We calculate the training and evaluation accuracy of different hyperparameters of RBF method and do analysis.

2 Data pre-processing

The data used in this project is Spam Data Set with 2000 training examples and 1536 testing examples and each example has a feature vector with 57 attributes. The labels of training examples and testing examples are denoted as -1 and +1 as non-spam and spam emails.

We standardize the data by removing the mean value of each feature and then dividing by each feature's standard deviation. We normalize the training and testing data with the equation (1). It helps center the values around the mean (u) with the standard deviation (σ).

$$X_{norm} = \frac{X - u}{\sigma} \tag{1}$$

The same procedure is taken for the evaluation datasets.

3 Compute discriminant function

3.1 Admissibility of the kernels

We compute the Gram matrix with Mercer's condition and calculate all the eigenvalues of the matrix. The matrix K contains some very small negative values. We set a very small negative value -10^{-4} as the threshold. As long as there is no eigenvalues smaller than it, then we believe the matrix is positive semi-definite and the kernel candidate is admissible which ensures that the SVM optimization problem is convex and has a unique global minimum.

We compute all the gram matrix and their eigenvalues and all of them are admissible.

3.2 A hard-margin SVM with the linear kernel $K(x_1, x_2) = x_1^T x_2$

The margin with the linear kernel can separate two classes of data if and only if there exist a hyper plane which w is the weight vector and the b is the bias. In order to find an optimal hyper plane, we can apply the dual problem and use the KKT conditions. The object of our task :

Find: α

Maximize: $Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j {x_i}^T x_j$

Subject to: $\sum_{i=1}^{N} \alpha_i \, d_i = 0, 0 \le \alpha_i \le C$

In theory, the value of C is $+\infty$, and in this task, we set the value C as 10^6. We use the *quadprog* function in MATLAB to solve this constraint optimization problem and get the α_i.

Based on (Karush-Kuhn-Tucker) KKT conditions, we choose the threshold 10^{-4} to determine the corresponding $\alpha_{\mathrm{i}} > 10^{-4}$ to the support vectors. In the training set, we get the 367 examples satisfy this condition.

Then we can calculate the discriminant function with equation

$$g(x) = {w_0}^T x + b_0 \tag{2}$$

where the weight and bias can be calculated by (3), (4), (5) and get the 57*1 vector of w_0 and $b_0 = -11.37$ as shown in table 3.1.

$$w_0 = \sum_{i=1}^{N} \alpha_i d_i x_i \tag{3}$$

$$b_{0,i} = \frac{1}{d^{(s)}} - {w_0}^T x^{(s)} \tag{4}$$

$$(x^{(s)} \; is \; a \; support \; vector \; with \; lable \; d^{(s)})$$

$$b_0 = \frac{\sum_{i=1}^{m} b_{0,i}}{m} \tag{5}$$

3.3 A hard-margin SVM with the polynomial kernel $K(x_1, x_2) = ({x_1}^T x_2 + 1)^P$

The hard-margin with a polynomial kernel $K(x_1, x_2) = ({x_1}^T x_2 + 1)^P$ which p range from 2 to 5. Then we apply the KKT condition and reduce unknowns to form the dual problem:

Find: α

Maximize: $Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j ({x_i}^T x_j + 1)^P$

Subject to: $\sum_{i=1}^{N} \alpha_i \, d_i = 0, 0 \le \alpha_i \le C$

We also set the C as 10^6. We solve the quadratic objective function to get the 2000 dimensions vector α with *quadprog* function in MATLAB. We select the support vectors with the value of α bigger than 10^{-4} with 174 examples.

In order to get the parameters including w_0 and b_0 in discriminant function (8), we

should compute as the equation (6) and (7) and do average calculation with (5)

$$w_0 = \sum_{i=1}^{N} \alpha_i d_i \varphi(x_i) \tag{6}$$

$$b_{0,i} = \frac{1}{d^{(s)}} - {w_0}^T \varphi(x^{(s)}) \tag{7}$$

The main issue of getting the parameters of w_0 and b_0 is without knowing $\varphi(\cdot)$, but it is difficult to find explicit form of $\varphi(\cdot)$. Then we can find expression for $K(\cdot,\cdot)$ directly. We can get the discriminant function without the function φ and find the value of bias with all the support vectors in the training set

$$g(x) = {w_0}^T \varphi(x) + b_0 = \sum_{i=1}^{N} \alpha_i d_i ({x_i}^T x + 1)^P + b_0 \tag{8}$$

The discriminative parameters are shown in table 3.1

3.4 A soft-margin SVM with the polynomial kernel $K(x_1, x_2) = ({x_1}^T x_2 + 1)^P$

All of other calculation steps are the same as 3.2 and the only difference is the boundary of α. Due to the difference of function to be minimized

$$f(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i \tag{8}$$

Value C reflects cost of violating constraints. A large C leads to smaller margin and fewer misclassification and a small C leads to larger margin and more misclassification of training data. According to the KKT condition and we can form the dual problem:

Find: α

Maximize: $Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j ({x_i}^T x_j + 1)^P$

Subject to: $\sum_{i=1}^{N} \alpha_i d_i = 0, 0 \le \alpha_i \le C$

The dual problem with soft margin is the same with 3.2 and the only difference is the upper limit of α. We tried to adapt the parameter of C and p and get different optimal boundary and discriminant function. The weight vector can't be calculated because we are without knowing the specific $\varphi(x)$ and calculate the kernel instead. We calculate the bias with the equation (8) and (5) and the results are shown in table 3.1.

Table 3.1 The parameters of discriminant function

Type of SVM	w_0				b_0			
Hard margin with linear kernel	57*1 matrix				-11.37			
Hard margin with polynomial kernel	p=2	p=3	p=4	p=5	p=2	p=3	p=4	p=5
	/	/	/	/	-18.89	-78.17	-439.50	1.83×10^{11}

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	/	/	/	/	-0.61	-1.25	-1.41	-1.46
p=2	/	/	/	/	-0.03	0.20	0.36	0.45
p=3	/	/	/	/	-0.16	-0.28	-0.29	-0.27
p=4	/	/	/	/	-0.36	-0.45	-0.70	-0.71
p=5	/	/	/	/	-0.45	-0.42	-0.37	-0.31

4 Training and testing accuracy calculation

4.1 A hard-margin SVM with the linear kernel $K(x_1, x_2) = x_1^T x_2$

We use the discriminant function calculated in part 3 and get the prediction of training and testing set according to the equation (2). To classify a new data point, we use the signum function defined

$$sgn[g(x_{test})] = \begin{cases} +1, & if\ g(x_{test}) \geq 0 \\ -1, & if\ g(x_{test}) \leq 0 \end{cases} \tag{9}$$

Then we compare the label of training and testing examples to calculated the accuracy with training accuracy of 93.95% and testing accuracy of 90.62% recorded in table 4.1.

4.2 A hard-margin SVM with the polynomial kernel $K(x_1, x_2) = (x_1^T x_2 + 1)^P$

Due to the polynomial kernel, we are without knowing the parameter of w_0 and we can't use the equation (2) to get the prediction of training and testing label. We use the equation (10) to do prediction. We also use the *sgn* function to do classification with equation (9).

$$g(x) = \sum_{i=1}^{N} \alpha_i d_i (x_i^T x + 1)^P + b_0 \tag{10}$$

When we change the parameter of p, we can get the training and testing accuracy with the comparison of their labels and the results are shown in table 4.1. A higher degree polynomial kernel model more complex decision boundary. However, using a very high degree polynomial kernel can also lead to overfitting which means the testing accuracy are reduced while the training accuracy are close to 100%. It is obvious that when the degree range from 4 to 5, the training and testing accuracy have only about 50% which means that the model is too complex and starts to fit the noise in the data instead of the underlying pattern. We choose degree 2 or 4 is suitable for this spam data set.

4.3 A soft-margin SVM with the polynomial kernel $K(x_1, x_2) = (x_1^T x_2 + 1)^P$

Compared with the hard-margin SVM, a soft-margin SVM have different upper boundary of α and the other calculations are the same. We calculated the training and testing accuracy with the same method mentioned in part 4.2 and the results are shown in table 4.1.

We modify the degree p of polynomial kernel. If we keep the same of parameter C to compare each row in the same column, it is very clear that with the increase of p, the

training accuracy is higher and higher while the testing accuracy is lower and lower. This condition indicates that when p=2, the testing accuracy is relative high and more suitable for this dataset.

We also try to modify the cost C of violating constraints to do comparison. We fix he degree p and compare each column in the same row. A large C leads to smaller margin and fewer misclassification which means the higher accuracy of training set. If the C is too large, the training accuracy will be high and testing accuracy will reduce and the over-fitting problem occurs. In this dataset, C=0.6 is a proper value for good classification performance.

Table 4.1 Accuracy of SVM classification

Type of SVM	Training accuracy				Test accuracy			
Hard margin with linear kernel	93.95%				90.62%			
Hard margin with polynomial kernel	p=2	p=3	p=4	p=5	p=2	p=3	p=4	p=5
	99.85%	99.90%	99.90%	49.75%	76.37%	71.81%	75.72%	49.67%
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	93.35%	93.85%	93.70%	93.90%	91.99%	92.19%	91.99%	92.25%
p=2	89.65%	99.10%	99.15%	99.30%	92.32%	91.80%	91.41%	91.34%
p=3	99.50%	99.70%	99.70%	99.70%	91.93%	91.28%	89.84%	88.93%
p=4	99.75%	99.75%	99.85%	99.85%	89.13%	86.98%	85.94%	85.09%
p=5	99.85%	99.85%	99.85%	99.90%	85.03%	84.38%	83.85%	84.64%

5 Radial Basis Function (RBF) kernel and implementation

RBF kernel is a type of non-linear kernel that maps data to a higher-dimensional feature space, where it can be more easily separable. The RBF kernel is defined as formular (11) of which the exponential is the squared Euclidean distance between two feature vectors and γ is a scalar that defines how much influence a single training example. In this data set, there are 57 features of each example.

$$K(x_1, x_2) = e^{-\gamma||x_1 - x_2||^2} \tag{11}$$

$$\gamma = \frac{1}{nfeatures * \sigma^2} = \frac{1}{57 * \sigma^2} \tag{12}$$

We form evaluation dataset of 600 examples from remaining training set and testing set. After the model trained by the 2000 training set, we use the evaluation dataset to do assessment. Through the same steps above, we can calculate the training and testing accuracy of different selection of hyperparameters C and σ.

The RBF kernel measures the similarity between two data points in the feature space. The hyperparameter γ controls the influence of the distance between the data points

on the kernel value. A smaller γ value will result in a wider kernel and a smoother decision boundary, while a larger γ value will result in a narrower kernel and a more complex decision boundary. It is obvious that the we fix the C and compare each row in the same column, the larger σ (the smaller γ), the lower training and evaluation accuracy because of the smoother decision boundary for more misclassification.

The parameter C controls the trade-off between maximizing the margin and minimizing the classification error. In this dataset, we can observe that the larger c, the higher training and evaluation accuracy which shows this dataset can be perfect separated with a narrow margin.

Table 4.1 Accuracy of RBF SVM classification

Type of SVM	Training accuracy				Evaluation accuracy			
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	92.15%	94.45%	95.35%	95.65%	92.50%	94.00%	94.83%	94.83%
σ=2	92.05%	93.20%	93.35%	93.95%	93.17%	93.00%	93.50%	93.67%
σ=3	91.50%	92.50%	92.80%	93.45%	92.50%	93.17%	92.50%	93.17%
σ=4	91.15%	92.20%	92.40%	92.55%	91.50%	92.83%	93.17%	93.33%
σ=5	90.70%	92.15%	92.20%	92.55%	91.00%	92.83%	93.00%	92.67%