National University of Singapore

School of Electrical and Computer Engineering

CA2 Project 1: SVM for Classification of Spam Email Messages

Student:
Wanry Lin

Matriculation Number:
A0000001X

Email address:

EE5904 Neural Network

May 10, 2023

Data preprocessing

SVM is sensitive to the scale of the input feature. For all tasks, I apply a standardization method to preprocess the input data. This involves subtracting the mean and dividing the standard deviation of the training data. As a result, I obtain an input that is in a "standard normal" distribution. The mathematical expression of my preprocessing algorithm is shown:

$$\textbf{standardization} : x^* = \frac{x - \overline{x}}{\sigma}$$

where:

$$\begin{aligned} \textbf{mean} : \overline{x} &= \frac{1}{N}\sum_{i}^{N} x_i \\ \textbf{standard deviation} : \sigma &= \sqrt{\frac{1}{N}\sum_{i}^{N}(x - \overline{x})^2} \end{aligned} \tag{1}$$

Mercer condition check

In all tasks, before doing the kernel mapping, I will do Mercers condition check to demonstrate admission of the kernel. It can be used as a kernel if it satisfies Mercers Condition, which can be expressed as:

$$\mathbf{K} = \begin{bmatrix} K(x_1,x_1) & K(x_1,x_2) & \dots & K(x_1,x_N) \\ \vdots & \vdots & \ddots & \vdots \\ K(x_N,x_1) & K(x_N,x_2) & \dots & K(x_N,x_N) \end{bmatrix} \in R^{N\times N}, \tag{2}$$

where N is the number of training examples and K should be positive semi definite. Threshold is 10^{-4} because numerical reason.

Task 1

In this task, I build and train the 3 kinds of SVM model by the training dataset. Here to test the kernel used in this task, I check the Mercer condition for every kernel. The result is shown in the following table. (P stands for pass and F stands for fail)

Table 1: Mercers Condition check for all kernels

Type of SVM	**Mercers Condition**			
Hard margin with linear kernel	P			
Hard margin with polynomial kernel	$p=2$	$p=3$	$p=4$	$p=5$
	P	P	F	F
Soft margin with polynomial kernel	$C=0.1$	$C=0.6$	$C=1.1$	$C=2.1$
$p=1$	P	P	P	P
$p=2$	P	P	P	P
$p=3$	P	P	P	P
$p=4$	F	F	F	F
$p=5$	F	F	F	F

From the Table 1, when $p=4 or 5$, the kernel is always failed in Mercer's condition check. However, for this polynomial kernel, the kernel matrix K is defined as $K(x_i,x_j)=(x_i^T x_j+1)^p$. It can be shown that K is positive semi-definite for all values of p, which means that the polynomial kernel satisfies Mercer's condition. But in practice,the value is so small that it is numerically equal to 0 for some values of p. Since the input feature has been standardized between 1 and 0, when p is too large (p=4 or p=5), the value exponential decreases to 0 (10^{-n})but never reach 0. Hence it cannot pass the Mercer's condition check if it takes value less than 10^{-4} equal to 0.

Task 2

The accuracy of training data and test data are denoted by:

$$\textbf{Accuracy} = \frac{\sum_{i=1}^{N}(pred_i = di)}{\sum_{i=1}^{N} d_i}$$

The detailed comparison of the kernels of SVM is listed in Table 2.

Table 2: Results of SVM classification

Type of SVM	**Training accuracy %**				**Testing accuracy %**			
Hard margin with linear kernel	93.95				92.77			
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	99.65	84.55	91.02	90.43	89.78	82.23
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	92.25	92.58	92.51	92.45
$p=2$	98.65	99.10	99.15	99.30	92.97	93.03	92.90	93.10
$p=3$	99.50	99.70	99.70	99.70	92.58	92.38	92.12	92.25
$p=4$	97.50	97.40	97.80	97.65	90.11	89.32	89.32	89.39
$p=5$	93.75	93.00	93.05	93.25	86.46	86.59	86.46	86.39

Comments

(1) The hard margin with linear kernel SVM achieves a relatively good performance with accuracy over 93% in training set and 92% in test set. It illustrates that more complex model does not equal to better performance.

(2) The hard margin with polynomial kernel SVM achieves various performance due to different p. For $p = 2 and p = 3$, the accuracy of training set is over 99% while the accuracy of test set is only around 91%. This result shows that SVM is overfitting. Meanwhile, though $p = 4$ is inadmissible kernel, the performance is relatively acceptable. It shows that Mercer's condition is just the sufficient condition for SVM classification not necessary condition. However, the performance for $p = 5$ is much worse than others.

(3) The soft margin with polynomial kernel SVM achieves various performance due to different p. For different C, the performance of SVM is close. In this task, the role of C is not obvious. But in practice, a small C gives little punishment for misclassification in training. Therefore, as C increases, the training accuracy grows too. However, it may lead to overfitting for the model and the accuracy in test set may drops. But it depends on the specific dataset.

(4) For $p = 1$, the result is similar to hard margin with linear kernel, since the kernel is close. For this p, the accuracy of training and test set are around 93%. For $p = 2 and p = 3$, the accuracy of training set is close 100% while the accuracy of test set is around 93%. There is overfitting for this kernel. For $p = 4 and p = 5$, the Mercer's condition points out that these kernel is inadmissible. The performance shows this conclusion.

Task 3

The Gaussian Radial Basis Function (RBF) kernel is a widely used kernel function in Support Vector Machines (SVMs) for classification and regression tasks. The RBF kernel is a type of kernel function that calculates the similarity between two points in a high-dimensional space. Specifically, the Gaussian RBF kernel calculates the similarity between two points, $x and x'$, based on the exponential of the negative of the squared Euclidean distance between them, multiplied by a hyperparameter γ:

$$K(x,x') = \exp(-\gamma * ||x - x'||^2)$$

The γ parameter determines the width of the kernel, controlling the smoothness of the decision boundary in the SVM. When gamma is small, the kernel is wide and the decision boundary is smooth, while larger values of gamma produce a narrower kernel and a more complex decision boundary. The Gaussian RBF kernel allows SVMs to model non-linear decision boundaries by mapping the input data into a higher-dimensional feature space. This enables SVMs to handle complex, non-linear relationships between the input features and the target variable.

Here, I use RBF kernel for my self-designed SVM model. The performance of RBF-SVM with different $\gamma and C$ is shown in the following table:

Table 3: Results of RBF SVM classification

Type of SVM	**Training accuracy %**					**Testing accuracy %**				
Soft margin	$C=1$	$C=10$	$C=100$	$C=316$	$C=1000$	$C=1$	$C=10$	$C=100$	$C=316$	$C=1000$
$\gamma=0.01$	90.05	93.00	95.80	98.15	99.65	88.99	91.79	93.81	95.18	95.16
$\gamma=0.0175$	93.40	96.75	99.75	99.8	99.85	92.12	94.40	95.89	95.89	95.70
$\gamma=0.1$	95.50	99.65	99.80	99.90	99.95	93.55	95.57	95.63	95.44	95.44
$\gamma=0.5$	98.55	99.80	99.90	99.95	100	94.79	94.92	94.98	94.85	94.92
$\gamma=1$	99.40	99.80	99.95	99.95	100	92.25	92.25	92.31	92.38	92.38

Table3 is a wide range and long step pre-searching for the best parameter. From Table3, it is obvious that the best γ is around 0.0175 between 0.01 and 1. The best C is around 100. Therefore, I make a short step search around $\gamma = 0.0175$ and $C = 100$, the result is shown in following table.

Table 4: Results of RBF SVM classification

Type of SVM	**Training accuracy %**						**Testing accuracy %**					
Soft margin	$C=70$	$C=85$	$C=100$	$C=115$	$C=130$	$C=150$	$C=70$	$C=85$	$C=100$	$C=115$	$C=130$	$C=150$
$\gamma=0.00175$	96.25	96.55	96.70	96.90	97.15	97.65	94.01	94.20	94.33	94.33	94.59	94.92
$\gamma=0.00877$	99.2	99.40	99.50	99.65	99.70	99.72	95.70	95.63	95.50	95.70	95.76	95.83
$\gamma=0.0175$	99.70	99.75	99.75	99.75	99.75	99.75	95.70	95.83	95.89	95.83	95.83	95.83
$\gamma=0.0263$	99.75	99.75	99.75	99.75	99.85	99.80	95.89	95.70	95.70	95.76	95.76	95.76
$\gamma=0.035$	99.75	99.75	99.85	99.80	99.80	99.80	95.57	995.70	95.76	95.76	95.70	95.70

From Table.4, the best performance is when $C = 100$ and $\gamma = 0.0175$.

Comments

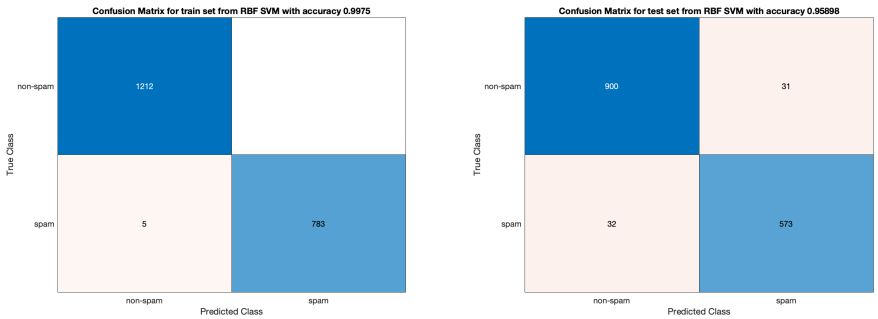
(1) The parameter C, also known as the penalty parameter, represents the tolerance for misclassification of samples. A higher value of C indicates a lower tolerance for misclassification, which can lead to overfitting. Conversely, a lower value of C results in higher tolerance for misclassification, which can lead to underfitting. Choosing an inappropriate value for C can lead to poor generalization performance of the model.

(2) "If γ is set too high, it can easily lead to overfitting. This is because the value of σ will be very small, resulting in a very narrow and tall Gaussian distribution. This distribution will only have an effect on the samples near the support vectors, resulting in poor classification performance for unknown samples. However, the training accuracy can be very high. If the value of γ is set to infinitesimal, theoretically, the Gaussian kernel SVM can fit any non-linear data, but it is prone to overfitting. On the other hand, if the γ is close to 0, the model prone to underfitting.

(3) Here, the value of γ is set from $\gamma = \frac{1}{n_{feature}}$, where $n_{feature}$ is the number of feature. This is the default measurement from sk-learn.

Performance of self-designed SVM

The confusion matrix of training set and test set are given.



(a) training set (b) test set

Figure 1: performance of self-designed SVM

Conclusion

From this project, I review the structure of SVM and learn the difference of linear kernel, polynomial kernel and the difference of hard margin, soft margin. It is not always the complex kernel the best. It depends on the task and dataset. Soft margin can effectively reduce overfitting and improve the performance in test set. I also learn the Mercer's condition is really useful for kernel design. In the 3rd task, I design a RBF kernel, since it is commonly used in machine learning. It is more powerful than linear and polynomial kernel. I find the best hyperparameter via learn the skill from sk-learn and test. In my opinion, this may derive to the model overfitting to test set and perform weak in evaluation set.