NYCU Introduction to Machine Learning, Homework 4

110550080, 何曉嫻

Part. 1, Coding (50%):

For this coding assignment, you are required to implement some fundamental parts of the Support Vector Machine Classifier using only NumPy. After that, train your model and tune the hyperparameter on the provided dataset and evaluate the performance on the testing data.

(50%) Support Vector Machine

1. (10%) Show the accuracy score of the testing data using *linear_kernel*. Your accuracy score should be higher than 0.8.

```
Accuracy of using linear kernel (C = 5):  0.83
```

2. (20%) Tune the hyperparameters of the *polynomial_kernel*. Show the accuracy score of the testing data using *polynomial_kernel* and the hyperparameters you used.

```
Accuracy of using polynomial kernel (C = 1, degree = 3):  0.98
```

3. (20%) Tune the hyperparameters of the *rbf_kernel*. Show the accuracy score of the testing data using *rbf_kernel* and the hyperparameters you used.

```
Accuracy of using rbf kernel (C = 1, gamma = 1):  0.99
```

Part. 2, Questions (50%):

1. (20%) Given a valid kernel k1(x,x'), prove that the following proposed functions are or are not valid kernels. If one is not a valid kernel, give an example of k(x,x') that the correspon ding K is not positive semidefinite and shows its eigenvalues.

    a. $k(x, x') = k_1(x, x') + exp(x^T x')$

    b. $k(x, x') = k_1(x, x') - 1$

    c. $k(x, x') = exp(\|x - x'\|^2)$

    d. $k(x, x') = exp(k_1(x, x')) - k_1(x, x')$

(a)

$k_2(x, x') = x^T x'$, if $\phi(x) = x$

→ $x^T x'$ is valid kernel

By 6.16 exp of v.k is v.k

→ $exp(x^T x')$ is valid kernel

By 6.17 addition of two v.k is v.k.

⇒ $k(x, x') = k_1(x, x') + exp(x^T, x')$ is valid kernel

(b)

$k_1(x, x') = x^T \cdot x'$ $x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ $x_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$

k_1 is kernel, $k_1(x_1, x_2) = 11$

$k(x_1, x_2) = 11 - 1 = 10$

Gram Matrix $K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) \\ k(x_2, x_1) & k(x_2, x_2) \end{bmatrix}$

$= \begin{bmatrix} 0 & 10 \\ 10 & 0 \end{bmatrix}$

eigenvalues of K are −10, 10 → not all non-negative

⇒ $k(x, x') = k_1(x, x') - 1$ is not valid kernel

(c)

Suppose $k_1(x, x') = x - x'$ $x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ $x_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$

$k(x_1, x_2) = \exp(8)$

Gram matrix $K = \begin{bmatrix} 1 & \exp(8) \\ \exp(8) & 1 \end{bmatrix}$

eigenvalues of K are $1 \pm \exp(8)$

$\rightarrow$ not positive semi-define

$\Rightarrow$ $k(x, x') = \exp(\|x - x'\|^2)$ is not valid kernel

(d)

$$\exp(k_1(x, x')) - \frac{k_1(x, x')}{1!}$$

$$= 1 + \frac{k_1(x, x')^2}{2!} + \cdots + \frac{k_1(x, x')^n}{n!}$$

By $\underline{6.18}$ and $\underline{6.16}$ $\frac{k_1(x, x')^t}{t!}$ is v.k

for $t = 2, 3 \cdots n$

1 is also v.k because the eigenvalues of

its gram matrix are 0, 1

By $\underline{6.17}$ $\exp(k_1(x, x')) - k_1(x, x')$ is v-k

$\Rightarrow$ $k(x, x') = \exp(k_1(x, x')) - k_1(x, x')$ is valid kernel

2. (15%) One way to construct kernels is to build them from simpler ones. Given three possib le "construction rules" : assuming K1(x, x') and K2(x, x')are kernels then so are
   a. (scaling) f(x)K1(x, x')f(x'), f(x)R
   b. (sum) K1(x, x')+K2(x, x')
   c. (product) K1(x, x')K2(x, x')

   Use the construction rules to build a normalized cubic polynomial kernel:

$$K(x, x') = \left(1 + \left(\frac{x}{||x||}\right)^T \left(\frac{x'}{||x'||}\right)\right)^3$$

   You can assume that you already have a constant kernel K0(x, x') = 1 and a linear kernel K1(x, x')=xTx'. Identify which rules you are employing at each step.

$$y = \frac{x}{||x||} \quad , \quad y' = \frac{x'}{||x'||} \qquad y^T = \frac{x^T}{||x||}$$

$$\begin{aligned} K(x, x') &= (1 + y^T y')^3 \\ &= (y^T y')^3 + 3(y^T y')^2 + 3(y^T y') + 1 \\ &= \underset{①}{\frac{(K_1(x, x'))^3}{||x||^3\ ||x'||^3}} + \underset{②}{\frac{3(K_1(x, x'))^2}{||x||^2\ ||x'||^2}} + \underset{③}{\frac{3(K_1(x, x'))}{||x||\ ||x'||}} + K_0(x, x') \end{aligned}$$

① By product rule, $(K_1(x, x'))^3$ is kernel

By scaling rule, $\underbrace{\frac{1}{||x||^3}}_{f(x)} (K_1(x, x'))^3 \underbrace{\frac{1}{||x'||^3}}_{f(x')}$ is kernel

② By product rule, $(K_1(x, x'))^2$ is kernel

By scaling rule, $\underbrace{\frac{\sqrt{3}}{||x||^2}}_{f(x)} (K_1(x, x'))^2 \underbrace{\frac{\sqrt{3}}{||x'||^2}}_{f(x')}$ is kernel

③ By scaling rule, $\underbrace{\frac{\sqrt{3}}{||x||}}_{f(x)} K_1(x, x') \underbrace{\frac{\sqrt{3}}{||x'||}}_{f(x')}$ is kernel

$\Rightarrow$ By sum rule, $K(x, x')$ is kernel

3. (15%) A social media platform has posts with text and images spanning multiple topics like news, entertainment, tech, etc. They want to categorize posts into these topics using SVMs. Discuss two multi-class SVM formulations: `One-versus-one` and `One-versus-the-rest` for this task.

a. The formulation of the method [how many classifiers are required]

For N classes, ovo required $\frac{N*(N-1)}{2}$ classifiers.

ovr required N-1 classifiers.

b. Key trade offs involved (such as complexity and robustness).

OvO:

1. Requires more classifiers for large N compared to OvR, may be computationally expensive for large numbers of classes, and need more training time
2. More robust.
3. Susceptible to tie votes when classes are closely related, leading to uncertainty in the prediction.

OvR:

1. Works well with large datasets and a large number of classes without excessive computational burden.
2. May lead to higher classification errors when classes overlap significantly.
3. Handling imbalanced datasets might be easier due to separate binary classifiers for each class.

c. If the platform has limited computing resources for the application in the inference phase and requires a faster method for the service, which method is better.

OvR method might be a better choice. As it requires less classifiers, the computational load is generally lower than OvO.

Each classifier is trained on a subset of the data, making it faster during prediction.