NYCU Introduction to Machine Learning, Homework 4

109700046, 侯均頲

The screenshot and the figures we provided below are just examples. **The results below are not guaranteed to be correct.** Please make sure your answers are clear and readable, or no points will be given. Please also remember to convert it to a pdf file before submission. You should use English to answer the questions. After reading this paragraph, you can delete this paragraph.

Part. 1, Coding (50%):

(50%) Support Vector Machine

Criteria:

1. (10%) Show the accuracy score of the testing data using *linear_kernel*.
```
Accuracy of using linear kernel (C = 3.33):  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 = 1.7, degree = 8):  0.97
```
3. (20%) Tune the hyperparameters of the *rbf_kernel*. Show the accuracy score of the testing
```
Accuracy of using rbf kernel (C = 0.2, gamma = 2):  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 corresponding K is not positive semidefinite and shows its eigenvalues.
   1. $k(x,x')=k_1(x,x')+\exp(x^T x')$

(a) $k(x,x') = \underbrace{k_1(x,x')}_{\text{kernel matrix } k1 \text{ must be postive semidefinite}} + \exp(x^T x')$

result kernel matrix

$$K = K_1 + \exp(x^T x')$$

lets see $x_1 = [1,2]^T$ $x_2 = [3,4]^T$

$$K_1 = \begin{bmatrix} 5 & 11 \\ 11 & 25 \end{bmatrix} \quad \exp(x^T x') = \begin{bmatrix} e^5 & e^{11} \\ e^{11} & e^{25} \end{bmatrix}$$

$$K = \begin{bmatrix} 5+e^5 & 11+e^{11} \\ 11+e^{11} & 25+e^{25} \end{bmatrix}$$

$\Rightarrow$ not all eigenvalues are non-negative

$\Rightarrow$ k might not be positive semidefinite

$\Rightarrow$ may not be valid kernel #

2. kx, x'= k1x, x'-1

(b) $k(x,x') = \underbrace{k_1(x,x')}_{\text{k1 must be positive semidefine}} - 1$

than $K = K1 \boxed{-} \boxed{I} \rightarrow$ ~~also posit~~ also positive semidefine

because K1 is positive semidefine, and substracting an identity matrix will not ~~a~~ affect the truth of positive semidefine,

k(x,x') is a valid kernel

~~f~~ from (6,17)

$k(x,x') = \underbrace{k_1(x,x')}_{k_1(x,x')} + \underbrace{k_2(x,x')}_{-I}$

3. kx, x'= exp || x-x' || 2

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

$\Rightarrow k(x, x') = \exp(+x^T x) \exp(+(x')^T x')$
$\times \exp(-2x^T x')$

~~because RBF is valid~~

from (6,15)(6,16)(6,13) $k(x,x') = \overset{(-1)}{c}\, k_1(x,x')$

$\cancel{k(x,x')}$ $k(x,x') = \exp(k_1(x,x'))$

$k(x,x') = f(x) k_1(x,x') f(x')$

$\Rightarrow k(x,x')$ is a

valid kernel

4. kx, x'= expk1x,x'-k1x, x'

d. $k(x,x') = \underbrace{\exp(k(x,x'))}_{a} \underbrace{- k_1(x,x')}_{b}$

from (6.16) $k(x,x') = \exp(k_1(x,x'))$

$\Rightarrow$ a is valid

b valid because $c \cdot k_1(x,x')$ is valid

from (6.17) $k(x,x')$

$= k_1(x,x') + k_2(x,x')$

$\Rightarrow$ a + b is valid

$\Rightarrow k(x,x')$ 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

1. (scaling) f(x)K1(x, x')f(x'), f(x)R
2. (sum) K1(x, x')+K2(x, x')
3. (product) K1(x, x')K2(x, x')
    1.

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

$$K(x, x')=1+x\|x\|Tx'\|x'\|3$$

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.

$$K(x,x') = \Big(1+\underbrace{\Big(\frac{x}{\|x\|}\Big)^T\Big(\frac{x'}{\|x'\|}\Big)}_{a}\Big)^3$$

lets say $f(x) = \frac{1}{\|x\|}$ $\quad f(x') = \frac{1}{\|x'\|}$

$\Rightarrow$ $a = f(x)\, k_1(x,x')\, f(x') = k_2(x,x')$

valid!

$\Rightarrow k(x,x') = (k_0(x,x') + k_2(x,x'))^3$

$= k_0(x,x')^3 + k_2(x,x')^3 + 3k_0(x,x')^2 k_2(x,x') + 3k_0(x,x')(k_2(x,x'))^2$ #

(by applying sum and product rule it is a valid kernel)

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

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

for n classes

One-versus-one: n * (n - 1) / 2

One-versus-the-rest: n

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

One-versus-one is more computationally expensive but can more rubust in the result, especially w hen classes are imbalanced, when the dataset is not too large we can consider this method, while O ne versus the rest should has less complexity and robustness but it can be more efficient than One-versus-one, it is more straightforward to implement and can be used in datasets with large numbers of classes.

3. 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.

As we can see from b., the answer should be One-versus-the-rest because it is more efficient and should has less complexity and be less computationally expensive.