NYCU Intro. to ML
HW4 Report

110550093, 蔡師睿

Part I. Coding

Support Vector Machine

1. Show the accuracy score of the testing data using *linear_kernel*.

```
Accuracy of using linear kernel (C = 4):  0.83
```

2. 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. 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 II. Questions

1. Given a valid kernel $k_1(x, x')$, prove that the following proposed are or are not valid kernels. If one is not a valid kernel, give an example of $k(x, x')$ that the corresponding $\boldsymbol{K}$ is not positive semidefinite and shows its eigenvalues.

$$\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= ck_1(\mathbf{x}, \mathbf{x}') && (6.13)\\
k(\mathbf{x}, \mathbf{x}') &= f(\mathbf{x})k_1(\mathbf{x}, \mathbf{x}')f(\mathbf{x}') && (6.14)\\
k(\mathbf{x}, \mathbf{x}') &= q\left(k_1(\mathbf{x}, \mathbf{x}')\right) && (6.15)\\
k(\mathbf{x}, \mathbf{x}') &= \exp\left(k_1(\mathbf{x}, \mathbf{x}')\right) && (6.16)\\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') && (6.17)\\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}') && (6.18)\\
k(\mathbf{x}, \mathbf{x}') &= k_3\left(\boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}')\right) && (6.19)\\
k(\mathbf{x}, \mathbf{x}') &= \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}' && (6.20)\\
k(\mathbf{x}, \mathbf{x}') &= k_a(\mathbf{x}_a, \mathbf{x}'_a) + k_b(\mathbf{x}_b, \mathbf{x}'_b) && (6.21)\\
k(\mathbf{x}, \mathbf{x}') &= k_a(\mathbf{x}_a, \mathbf{x}'_a)k_b(\mathbf{x}_b, \mathbf{x}'_b) && (6.22)
\end{aligned}$$

Figure 1: Known valid kernels

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

First, assume the kernel function $k_2(x, x') = x^T x'$, and suppose the feature space $\phi(x) = x$. Therefore, we get the derivation: $k_2(x, x') = x^T x' = \phi(x)^T \phi(x')$, which represents that $k_2(x, x')$ is a valid kernel.
Next, due to the valid kernel (6.16), $exp(x^T x')$ is also a valid kernel. According to rule (6.17), the addition of two valid kernels results in another valid kernel, and thus, we can prove that $k(x, x') = k_1(x, x') + exp(x^T x')$ is a valid kernel.

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

Suppose $x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $x' = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$, and $k_1(x, x') = \begin{bmatrix} 1 & 3 \\ 3 & 13 \end{bmatrix}$, so $k(x, x') = \begin{bmatrix} 1 & 3 \\ 3 & 13 \end{bmatrix} - 1 = \begin{bmatrix} 0 & 2 \\ 2 & 12 \end{bmatrix} = \boldsymbol{K}$, where $\boldsymbol{K}$ is the corresponding. The eigenvalues of $\boldsymbol{K}$ are obtained from $det(\boldsymbol{K} - \lambda\mathbf{I})$, resulting in $\lambda = 6 \pm 2\sqrt{10}$. Since there is an negative eigenvalue $(6 - 2\sqrt{10} < 0)$, the corresponding $\boldsymbol{K}$ isn't positive semidefinite and $k(x, x') = k_1(x, x') - 1$ isn't a valid kernel.

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

Suppose $x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $x' = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$, so $k(x, x') = \begin{bmatrix} 1 & exp(8) \\ exp(8) & 1 \end{bmatrix} = \boldsymbol{K}$, where $\boldsymbol{K}$ is the corresponding. The eigenvalues of $\boldsymbol{K}$ are obtained from $det(\boldsymbol{K} - \lambda\mathbf{I})$, resulting in $\lambda = 1 \pm exp(8)$. Since there is an negative eigenvalue $(1 - exp(8) < 0)$, the corresponding $\boldsymbol{K}$ isn't positive semidefinite and $k(x, x') = exp(\|x - x'\|^2)$ isn't a valid kernel.

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

To simplify the problem, we can use Taylor expansion to expand $exp(k_1(x, x'))$, and then we get $k(x, x') = 1 + k_1(x, x') + \frac{k_1(x,x')^2}{2!} + \frac{k_1(x,x')^3}{3!} + \cdots - k_1(x, x') = 1 + \frac{k_1(x,x')^2}{2!} + \frac{k_1(x,x')^3}{3!} + \cdots = 1 + \sum_{n=2}^{\infty} \frac{k_1(x,x')^n}{n!}$. We know that constant 1 and polynomial with non-negative coefficients $\frac{k_1(x,x')^n}{n!}$ are valid kernels, and according to rule (6.17), the addition of valid kernels results in another valid kernel. Therefore, we can prove that $k(x, x') = exp(k_1(x, x')) - k_1(x, x')$ is a valid kernel.

2. One way to construct kernels is to build them from simpler ones. Given three possible "construction rules": assuming $K_1(x, x')$ and $K_2(x, x')$ are kernels then so are
a. (scaling) $f(x)K_1(x, x')f(x'), f(x) \in R$
b. (sum) $K_1(x, x') + K_2(x, x')$
c. (product) $K_1(x, x')K_2(x, x')$
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 $K_0(x, x') = 1$ and a linear kernel $K_1(x, x') = x^T x'$. Identify which rules you are employing at each step.

First, apply the scaling rule to linear kernel $K_1(x, x') = x^T x'$ with $f(x) = \frac{1}{\|x\|}$:

$$\boldsymbol{K_2}(x, x') = f(x)\boldsymbol{K_1}(x, x')f(x') = \frac{1}{\|x\|} x^T x' \frac{1}{\|x'\|} = \left(\frac{x}{\|x\|}\right)^T \left(\frac{x'}{\|x'\|}\right)$$

Second, apply the sum rule to add $K_0(x, x') = 1$ and $K_2(x, x')$:

$$\boldsymbol{K_3}(x, x') = \boldsymbol{K_0}(x, x') + \boldsymbol{K_2}(x, x') = 1 + \left(\frac{x}{\|x\|}\right)^T \left(\frac{x'}{\|x'\|}\right)$$

Finally, apply the product rule to construct a cubic term for K_3:

$$\boldsymbol{K_4}(x, x') = (\boldsymbol{K_3}(x, x'))^3 = \left(1 + \left(\frac{x}{\|x\|}\right)^T \left(\frac{x'}{\|x'\|}\right)\right)^3 = \boldsymbol{K}(x, x')$$

Therefore, we construct the normalized cubic polynomial kernel K successfully.

3. 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-rest' for this task.

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

In the 'One-versus-one' approach, the SVM is trained for each pair of two classes. Therefore, we will need totally $\frac{K(K-1)}{2}$ classifiers to train, where K is the number of classes. On the other hand, in the 'One-versus-rest' approach, the SVM is trained using data from a specific class as the positive examples and the data from the remaining classes as the negative data, and thus it only needs to construct K two-class SVM classifiers, where K is the number of classes.

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

Complexity : The strategy 'One-versus-one' will be more complex than 'One-versus-rest' as the number of classes is large, because 'One-versus-one' needs more SVM classifiers and costs more computational resource.

Robustness : 'One-versus-one' is more robust than 'one-versus-rest', because 'One-versus-one' is trained on two different classes at a time, each distinguishing between a pair of classes, and it is less sensitive to imbalances in class distribution.

In summary, there is a trade-off between complexity and robustness.

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.

Given the constraints of limited computing resources and the imperative for a faster approach, 'One-versus-rest' emerges as a more suitable strategy for the reasons outlined earlier, even though it requires sacrificing some robustness.