NYCU Introduction to Machine Learning, Homework 4

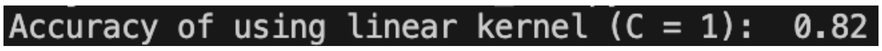
Deadline: Dec. 19, 23:59

109612019 林伯偉

Part. 1, Coding (50%):

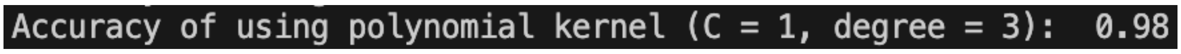
(50%) Support Vector Machine

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



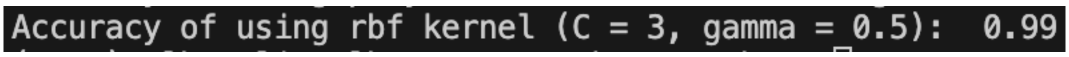
```
Accuracy of using linear kernel (C = 1):  0.82
```

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 = 3, gamma = 0.5):  0.99
```

(next page)

The following table is the grading criteria for question 2 and 3:

Points	**Testing Accuracy**
20 points	**0.98 <= acc**
15 points	**0.90 <= acc < 98**
10 points	**0.85 <= acc < 0.90**
5 points	**0.8 <= acc < 0.85**
0 points	**acc < 0.8**

Part. 2, Questions (50%):

1. (20%) Given a valid kernel $k_1(x, x')$, 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.
   a. $k(x, x') = k_1(x, x') + exp\,(x^T x')$
   b. $k(x, x') = k_1(x, x') - 1$
   c. $k(x, x') = exp\left(\|x - x'\|^2\right)$
   d. $k(x, x') = exp\left(k_1(x, x')\right) - k_1(x, x')$

(a)

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

Use the property that $k(x, x') = k_1(x, x') + k_2(x, x')$ is a valid kernal, if both $k_1(x, x')$ and $k_2(x, x')$ are valid kernals.

Using $x^T A x'$ is a valid kernal (6.20), prove I is a SPSM

$u^T I u = u^T u = \|u\|^2 \geq 0, \forall u \neq \vec{0}$, $\therefore$ I is a SPSM

$\Rightarrow x^T x'$ is a valid kernel (when $A = I$)

$\Rightarrow \exp(x^T x')$ is a valid kernal (6.16)

$$\begin{cases} k_1(x, x') = x^T x' \text{ is valid} \\ \exp(k_1(x^T, x')) \text{ is valid (from 6.16)} \end{cases}$$

So that $k(x.x')$ is a valid kernal $_{\#}$, Hence the proof

(b)

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

Suppose $K_1 = \begin{bmatrix} k_1(x_1, x_1) & k_1(x_2, x_1) \\ k_1(x_1, x_2) & k_1(x_2, x_2) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$

$\lambda_1 = 1, \lambda = 0$

Then $K = \begin{bmatrix} 1-1 & 0-1 \\ 0-1 & 0-1 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ -1 & -1 \end{bmatrix}$, $\lambda_1 = \frac{-\sqrt{5}-1}{2}$, $\lambda_2 = \frac{\sqrt{5}-1}{2}$

Since $\lambda_1 < 0$, so K is not a SPSM

Hence $k(x, x')$ is not a valid kernel $_{\#}$

(c)

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

$$\Rightarrow \|x - x'\|^2 = x^T x + (x')^T x' - 2x^T x'$$

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

$$= \underbrace{\exp(x^T x)}_{f(x)} \cdot \exp(-2x^T x') \cdot \underbrace{\exp(x'^T x')}_{f(x')}$$

Using 6.14 as long as $\exp(-2x^T x')$ is a SPSM

Since $\exp(-2x^T x)$ always > 0

$\therefore$ $u^T \exp(-2x^T x) u > 0 \quad \forall\, u \neq \vec{0}$

$\Rightarrow$ $\exp(-2x^T x)$ is a SPSM

$\therefore$ using 6.14, $k(x, x') = \exp(\|x - x'\|^2)$ is a valid kernel //

(d)

$$k(x, x') = \exp(k_1(x, x')) - k_1(x, x')$$

If $k_1 = \begin{bmatrix} x_1^T x_1 & x_1^T x_2 \\ x_2^T x_1 & x_2^T x_2 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is a SPSM

$\Rightarrow$ $\begin{vmatrix} a - \lambda & b \\ c & d - \lambda \end{vmatrix} = 0$ has all non-negative roots

$\Rightarrow$ $\lambda^2 - (a + d)\lambda + (ad - bc) = 0$

$\Rightarrow$ $\underset{(\lambda_1 + \lambda_2)}{a + d \geq 0}$, $\underset{(\lambda_1 \lambda_2)}{ad - bc \geq 0}$

$$k = \begin{bmatrix} e^a - a & e^b - b \\ e^c - c & e^d - d \end{bmatrix}$$

$$\Rightarrow \begin{vmatrix} e^a - a - \lambda & e^b - d \\ e^c - c & e^d - d - \lambda \end{vmatrix} = 0$$

$$\Rightarrow \lambda^2 - \underbrace{(e^a + e^d - a - c)}_{\text{should} \geq 0\ (\lambda_1 + \lambda_2)}\lambda + \underbrace{(e^{ad} - de^a - a^{ed} + ad)}_{\text{should} \geq 0\ (\lambda_1 \lambda_2)} = 0$$

$\therefore$ λ_1, λ_2 are both non-negative (SPSM)

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

2. (15%) 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.

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

$\frac{x^T x'}{\|x\|\|x'\|}$ is a scaling of $k_1(x, x') = x^T x$, so it's a valid kernel $(k_2(x, x'))$.

$1 + \frac{x^T x'}{\|x\|\|x'\|}$ is the sum of $k_0(x, x') = 1$ and $k_2(x, x')$ so it's a valid kernel $(k_3(x, x'))$.

$\left(1 + \frac{x^T x'}{\|x\|\|x'\|}\right)^3$ is the product of $3 k_3(x, x')$ so it's a valid 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]

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

        In one-versus-one, a binary classifier is trained for each pair of classes. For k classes, there are k(k−1)/2 classifiers. During inference, each classifier predicts the class, and the class with the most votes is chosen as the final prediction.

      - **One-versus-the-rest** : k

        In one-versus-the-rest (also known as one-versus-all), k binary classifiers are trained, each distinguishing between one specific class and the rest of the classes. During inference, the class associated with the classifier that outputs the highest confidence score is predicted.

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

      - **One-versus-one**

        Advantages:

        - Less affected by class imbalance issues.
        - Can handle non-linear decision boundaries effectively.

        Disadvantages:

        - Requires k(k-1)/2 classifiers, which can be computationally expensive for a large number of classes.
        - Training time increases quadratically with the number of classes.

      - **One-versus-the-rest:**

        Advantages:

        - Simplicity and ease of implementation.
        - Suitable for scenarios with class imbalance.
        - Training time is linear with the number of classes.

Disadvantages:

- May suffer from class imbalance issues.
- The class might influence decision boundaries with a larger number of instances.

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.

If the platform has limited computing resources for the application in the inference phase and requires a faster method, the **One-versus-the-rest** method is typically a better choice. Because:

- **Lower Computational Cost:**
  One-versus-the-rest requires fewer classifiers (linear with the number of classes), leading to lower computational costs during both training and inference.

- **Simplicity:**
  The simplicity of the one-versus-the-rest approach makes it computationally more efficient, especially when resources are limited.

- **Linear Training Time:**
  The training time scales linearly with the number of classes, making it more suitable for scenarios with limited resources.

While One-versus-one has its advantages, the quadratic increase in the number of classifiers may pose challenges in scenarios where computational resources are constrained. Therefore, for faster inference with limited computing resources, **One-versus-the-rest** is a better choice.