NYCU Introduction to Machine Learning, Homework 4

110550085 房天越

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 = 0.11451448763):  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 = 0.11451448763, degree = 3):  0.99
```

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 = 1.1451448763, gamma = 1):  0.99
```

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

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

      a. This is valid, by 6.20, $x^T x'$ is valid, then by 6.16, $\exp(x^T x')$ is valid, finally, by 6.17, $k_1(x, x') + \exp(x^T x')$ is valid, so this is valid.

      b. This is not valid, let $k_1(x, x')$ = have the kernel matrix K = [0.8 0.2, 0.2 0.8], then k(x, x') has the kernel matrix K' = [-0.2 -0.8, -0.8 -0.2], the eigenvalues are 0.6 or -1, which are not all nonnegative, so it does not satisfy the property of positive semi-definite, which implies that k(x, x') is not a valid kernel.

c. This is valid, because the exponential of any real number is positive, and the square of the Euclidean distance is always nonnegative.

d. This is not valid, let $k_1(x, x') = x^T x' + 1$, then the kernel matrix of the kernel function k(x, x') would be [e-1 e^2-2, e^2-2 e-1], the eigenvalues of this matrix would be one positive and one negative, so that would not be valid.

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.

At first, we have $K_1(x, x') = x^T x'$, then by scaling, let f(x) = 1/||x||, f(x') = 1/||x'||. So $K_2 = (x/||x||)^T(x'/||x'||)$ is valid, then by sum rule, K_2+K_0 is valid. Finally, by product rule, $K(x, x') = (K_2+K_0)\ (K_2+K_0)\ (K_2+K_0)$, the normalized cubic polynomial kernel is constructed.

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]
    b. Key trade offs involved (such as 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.

a. For One-versus-one formulation, a binary classifier is trained for each pair of classes, if there are N classes, then N(N-1)/2 binary classifiers are trained.
   And for One-versus-the rest formulation, a binary classifier is trained for each class against the rest of the classes, if there are N classes, N binary classifiers are trained.
b. For One-versus-one formulation, training is computationally expensive because it requires to train for each pair of classes, but it is more robust when dealing with imbalanced datasets because each binary classifier focuses only on distinguishing between two classes.
   And for One-versus-the-rest formulation, it is less intensive than one-versus-one when the number of classes is large, it could simplify the problem and is often more scalable. However, One-versus-the-rest is less robust because it is sensitive to class imbalances because it focuses on distinguishing each class from the rest. If a class has significantly fewer instances, the classifier may become biased towards the majority class.
c. Like what I mentioned in b., One-versus-the-rest might be a best choice in this situation. It is less computationally expensive as it requires fewer classifiers, and could provide a better service.