NYCU Introduction to Machine Learning, Homework 4

Deadline: Dec. 19, 23:59

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

Requirements:

- Implement the *gram_matrix* function to compute the Gram matrix of the given data with an argument **kernel_function** to specify which kernel function to use.
- Implement the *linear_kernel* function to compute the value of the linear kernel between two vectors.
- Implement the *polynomial_kernel* function to compute the value of the polynomial kernel between two vectors with an argument **degree**.
- Implement the *rbf_kernel* function to compute the value of the rbf kernel between two vectors with an argument **gamma**.

Tips:

- Your functions will be used in the SVM classifier from scikit-learn like the code below.
```
svc = SVC(kernel='precomputed')
svc.fit(gram_matrix(X_train, X_train, your_kernel), y_train)
y_pred = svc.predict(gram_matrix(X_test, X_train, your_kernel))
```
- For hyperparameter tuning, you can use any third party library's algorithm to automatically find the best hyperparameter, such as GridSearch. In your submission, just give the best hyperparameter you used and do not import any additional libraries/packages.

Criteria:

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

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

(next page)

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.

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.