NCTU Introduction to Machine Learning, Homework 4

Deadline: Nov. 29, 23:59

109550064 陳姵帆

Part. 1, Coding (50%):

In this coding assignment, you need to implement the cross-validation and grid search using only NumPy, then train the SVM model from scikit-learn on the provided dataset and test the performance with testing data. Find the sample code and data on the GitHub page https://github.com/NCTU-VRDL/CS_AT0828/tree/main/HW4

Please note that only NumPy can be used to implement cross-validation and grid search. You will get no points by simply calling sklearn.model_selection.GridSearchCV.

1. (10%) K-fold data partition: Implement the K-fold cross-validation function. Your function should take K as an argument and return a list of lists (*len(list) should equal to K*), which contains K elements. Each element is a list containing two parts, the first part contains the index of all training folds (index_x_train, index_y_train), e.g., Fold 2 to Fold 5 in split 1. The second part contains the index of the validation fold, e.g., Fold 1 in split 1 (index_x_val, index_y_val)

```
X = np.arange(20)
kfold_data = cross_validation(X, None, k=5) # k = 10 (original)
✓ 0.2s
```

```
for i in range(len(kfold_data)):
    print("Split: %s, Training index: %s, Validation index: %s" % (i+1, kfold_data[i][0], kfold_data[i][1]))
✓ 0.2s

Split: 1, Training index: [ 0  1  2  3  4  5  6  7  8  9 10 14 15 16 18 19], Validation index: [11 12 13 17]
Split: 2, Training index: [ 1  3  4  5  6  7  9 10 11 12 13 14 15 17 18 19], Validation index: [ 0  2  8 16]
Split: 3, Training index: [ 0  1  2  3  6  7  8 10 11 12 13 14 16 17 18 19], Validation index: [ 4  5  9 15]
Split: 4, Training index: [ 0  1  2  3  4  5  6  8  9 10 11 12 13 15 16 17], Validation index: [ 7 14 18 19]
Split: 5, Training index: [ 0  2  4  5  7  8  9 11 12 13 14 15 16 17 18 19], Validation index: [ 1  3  6 10]
```

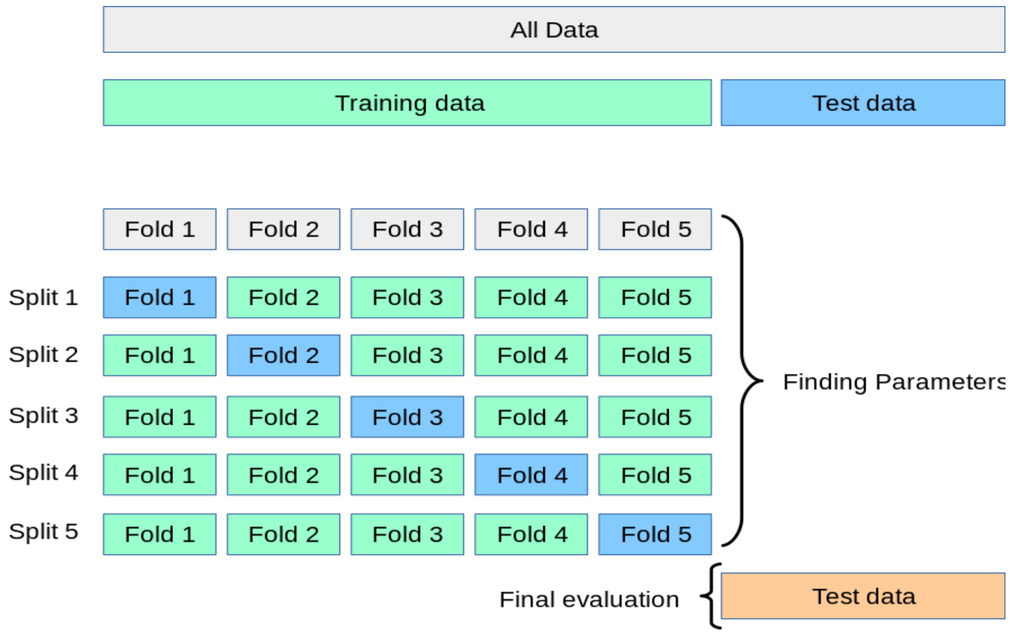
```
X = np.arange(11)
kfold_data = cross_validation(X, None, k=4) # k = 10 (original)
✓ 0.2s
```

```
for i in range(len(kfold_data)):
    print("Split: %s, Training index: %s, Validation index: %s" % (i+1, kfold_data[i][0], kfold_data[i][1]))
✓ 0.5s

Split: 1, Training index: [ 1  2  3  5  6  7  9 10], Validation index: [0 4 8]
Split: 2, Training index: [ 0  2  3  4  5  7  8 10], Validation index: [1 6 9]
Split: 3, Training index: [0 1 3 4 5 6 8 9], Validation index: [ 2  7 10]
Split: 4, Training index: [ 0  1  2  4  6  7  8  9 10], Validation index: [3 5]
```

Note: You need to handle if the sample size is not divisible by K. Using the strategy from sklearn. The first n_samples % n_splits folds have size n_samples // n_splits + 1, other folds have size n_samples // n_splits, where n_samples is the number of

samples, n_splits is K, % stands for modulus, // stands for integer division. See this [post](#) for more details
Note: Each of the samples should be used **exactly once** as the validation data
Note: Please **shuffle** your data before partition

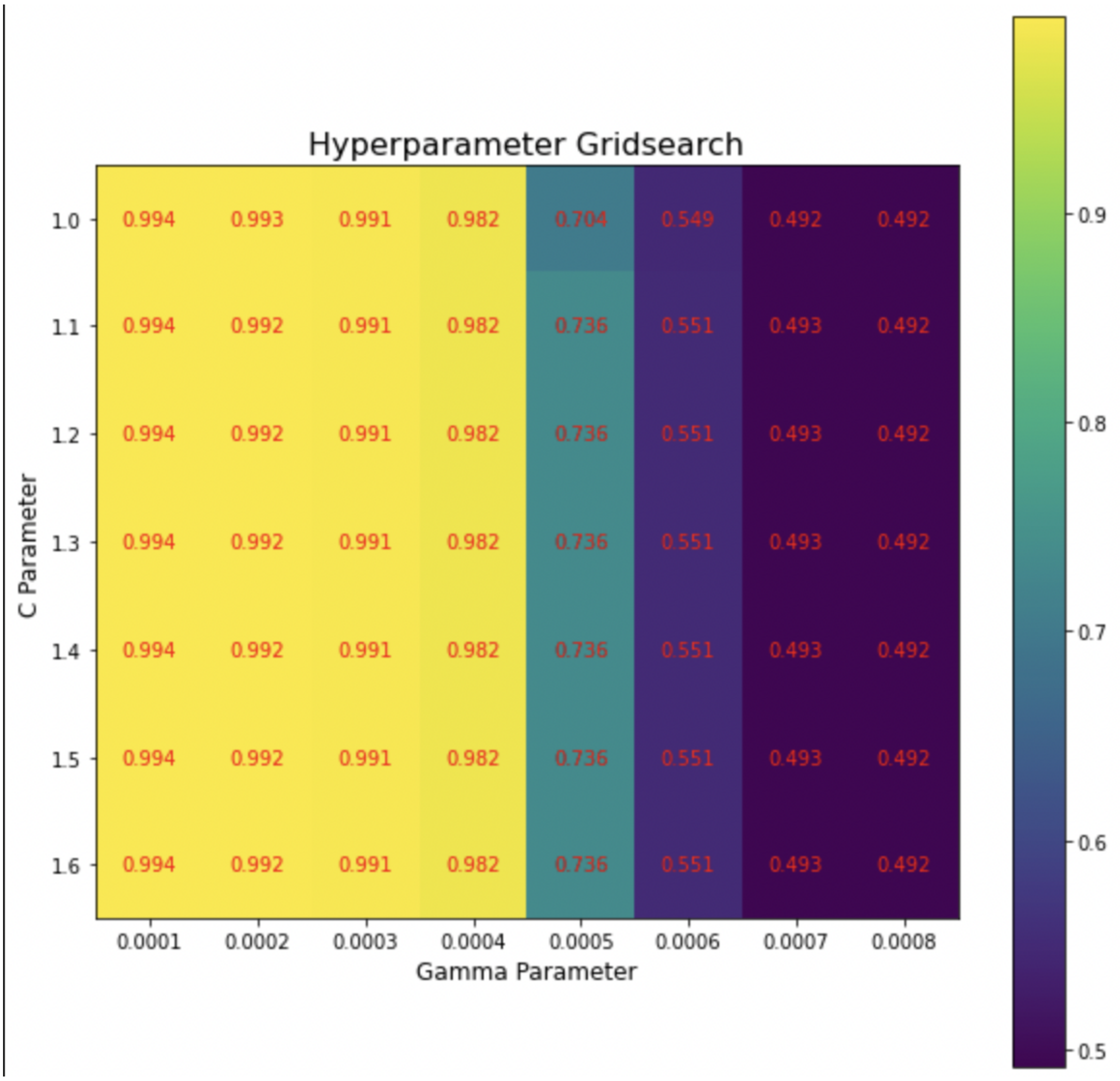


2. (20%) Grid Search & Cross-validation: using [sklearn.svm.SVC](#) to train a classifier on the provided train set and conduct the grid search of “C” and “gamma,” “kernel’=’rbf’ to find the best hyperparameters by cross-validation. Print the best hyperparameters you found.
   Note: We suggest using K=5

```
# print the best parameters I find
print("best parameters: c: ", best_c, " , gamma: ",best_gamma)
✓ 0.2s

best parameters: c:  1.0  , gamma:  0.0001
```

3. (10%) Plot the grid search results of your SVM. The x and y represent “gamma” and “C” hyperparameters, respectively. And the color represents the average score of validation folds.
   Note: [matplotlib](#) is allowed to use



4. (10%) Train your SVM model by the best hyperparameters you found from question 2 on the whole training data and evaluate the performance on the test set.

```
best_model = SVC(C=best_c, kernel='rbf', gamma=best_gamma)
best_model.fit(x_train, y_train)
✓ 5.9s

SVC(gamma=0.0001)
```

Accuracy	**Your scores**
acc > 0.9	**10points**
0.85 <= acc <= 0.9	**5 points**
acc < 0.85	**0 points**

Part. 2, Questions (50%):

(10%) Show that the kernel matrix $K = \left[k\left(x_n, x_m\right)\right]_{nm}$ should be positive semidefinite is the necessary and sufficient condition for $k(x, x')$ to be a valid kernel.

①

(1) kernel matrix $K = [k(x_n, x_m)]_{nm}$ positive semidefinite $\rightarrow$ $k(x, x')$ to be valid kernel

The nessary and sufficient condition for $k(x, x')$ to be a valid kernel:

Gram matrix $K = [k(x_n, x_m)]_{nm}$ should be positive semidefinite

for all possible choices of the set

- A matrix is positive semidefinite means that all of its eigen values are non-negative.
- K is symmetric. Thus, we have $K = V \Lambda V^T$
  - → where V is an orthonormal matrix V_t and the diagonal matrix Λ contains the eigenvalues λ_t of K
  - → if K is positive semidefinite, all eigenvalues are non-negative
  - → consider the feature map: $\phi : x_i \mapsto (\sqrt{\lambda_t} V_{ti})_{t=1}^{n} \in R^n$
  - → we find that

$$\phi(x_i)^T \phi(x_j) = \sum_{t=1}^{n} \lambda_t V_{ti} V_{tj} = (V \Lambda V^T)_{ij} = K_{ij} = k(x_i, x_j)$$

(2) $k(x, x')$ to be valid kernel $\rightarrow$ kernel matrix $K = [k(x_n, x_m)]_{nm}$ positive semidefinite

since $k(x, x')$ is valid kernel, it has a corresponding feature map ϕ such that $k(x, x') = \phi(x)^T \phi(x')$.

Thus, the kernel matrix K has entries $K_{ij} = \phi(x_i)^T \phi(x_j)$

Let V be the matrix $[\phi(x_1) \cdots \phi(x_n)]$, where we treat $\phi(x_i)$ as column vector. Then, we have $K = V^T V$.

However, this shows that K must be positive semidefinite because for any vector $z = R^{|S|}$ (if K is $S \times S$ dimension)

(10%) Given a valid kernel $k_1(x, x')$, explain that $k(x,\ x') = \ exp(k_1(x, x'))$ is also a valid kernel. Your answer may mention some terms like ____ series or ____ expansion.

② Using Taylor expansion around 0:

$$exp(K) = exp(0) + exp(0)K + \frac{exp(0)}{2!}K^2 + \frac{exp(0)}{3!}K^3 + \cdots$$

$$= 1 + K + \frac{1}{2}K^2 + \frac{1}{6}K^3 + \cdots$$

we can see that the exponential of a kernel is just an infinite series of multiplications and additions of that kernel.

Using the fact that addition and multiplication of valid kernels yield valid kernels:

$$K' = \alpha k_1 + \beta K_2 \quad (\alpha k_1 + \beta k_1)$$

$$K' = K_1 K_2 \quad (k_1 k_1)$$

we can conclude that the exponential of a kernel is a kernel.

* 上圖中的 alpha, aeta 都是大於 0 的 constant,
k(x, x’) = c*k1(x, x’), if c > 0 is a constant, k(x, x’) is valid (slide ch6 p15),
所以 alpha*k1 is valid kernel, beta*k1 is also valid kernel
(alpha*k1) + (beta*k1) still valid kernel

(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. $k(x,\ x') = \ k_1(x,\ x') + \ 1$

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

c. $k(x,\ x') = \ k_1(x, x')^2 + \ exp(\|x\|^2) * \ exp(\|x'\|^2)$

d. $k(x,\ x') = \ k_1(x, x')^2 + \ exp(k_1(x, x')) - \ 1$

③

a. <u>valid kernel</u>

given that if $k_1(x, x')$ is valid kernel, $k(x, x') = q(k_1(x, x'))$ is still a valid kernel if $q(\cdot)$ is a polynomial with nonnegative coefficient.

$k(x, x') = k_1(x, x') + 1 = q(k_1(x, x'))$,

where $q(x) = x + 1$, q is a polynomial with nonnegative coefficient

Thus, $k_1(x, x') + 1$ is valid kernel

b. <u>invalid</u>

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

$= \phi(x)^T \phi(x') - 1$

$= \underline{(x^T x')^2} - 1$ → (valid kernel function in slide ch6 p.13)

suppose $k(x, x')$'s K matrix :

$$k(x, x') = (x_1 x'_1 + x_2 x'_2)^2 - 1$$

$$K \text{ matrix} = \begin{bmatrix} \phi(x)^T \phi(x) - 1 & \phi(x)^T \phi(x') - 1 \\ \phi(x')^T \phi(x) - 1 & \phi(x')^T \phi(x')^T - 1 \end{bmatrix}$$

$$= \begin{bmatrix} (x_1^2+x_2^2)^2-1 & (x_1x_1'+x_2x_2')^2-1 \\ (x_1x_1'+x_2x_2')^2-1 & (x_1'^2+x_2'^2)^2-1 \end{bmatrix} = K$$

求 K 的 eigenvalue

$$A-\lambda I = \begin{bmatrix} (x_1^2+x_2^2)^2-1-\lambda & (x_1x_1'+x_2x_2')^2-1 \\ (x_1x_1'+x_2x_2')^2-1 & (x_1'^2+x_2'^2)^2-1-\lambda \end{bmatrix}$$

$$\det(A-\lambda I) = 0$$

$$= \left[(x_1^2+x_2^2)^2-1-\lambda\right]\left[(x_1'^2+x_2'^2)^2-1-\lambda\right]-\left[(x_1x_1'+x_2x_2')^2-1\right]^2=0$$

0.01

if $x=(x_1, x_2) = (1, -1)$ $\quad 0+0.1$

$x'=(x_1', x_2') = (0, 0.1)$ $\quad 0.01-1$

0.0001

$$[3-\lambda][-0.9999-\lambda] = [-0.99]^2$$

$$\lambda^2+(-3+0.9999)\lambda-3\times0.9999-0.99^2=0$$

$$\lambda^2-2.0001\lambda-2.9997-0.9801=0$$

$$\lambda^2-2.0001\lambda-3.9798=0$$

$$\lambda = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

$$= \frac{2.0001\pm\sqrt{2.0001^2-4\cdot1\cdot(-3.9798)}}{2\cdot1}$$

當

$$\Rightarrow \lambda = \frac{2.0001-\sqrt{2.0001^2+4\cdot3.9798}}{2} < 0$$

找到負數的 eigenvalue 了

$\Rightarrow$ invalid

c. valid

$k_1(x, x')^2 + \exp(\|x\|^2) * \exp(\|x'\|^2)$

let $k_1(x, x') = \phi(x)^T \phi(x')$

we can find new mapping function for $k(x, x')$

$$\phi'(x) = \begin{bmatrix} \phi(x) \\ \exp(\|x\|^2) \end{bmatrix}, \quad \phi'(x') = \begin{bmatrix} \phi(x') \\ \exp(\|x'\|^2) \end{bmatrix}$$

Thus it is valid kernel

d. valid

$$k(x, x') = k_1(x, x')^2 + \exp(k_1(x, x')) - 1$$

$$= k_1(x, x')^2 + [\underbrace{\cancel{1} + k_1(x, x') + \tfrac{1}{2} k_1(x, x')^2 + \cdots}_{\text{Taylor Expansion}}] \cancel{-1}$$

$$= \underbrace{\underbrace{k_1(x, x')^2}_{} + k_1(x, x') + \underbrace{\tfrac{1}{2} k_1(x, x')^2}_{} + \underbrace{\cdots}_{}}_{\text{Addition of the valid kernel}}$$

multiplication of valid kernel

Addition of the valid kernel

$\Downarrow$

$k(x, x')$ is valid kernel

*All coefficients are positive in the Taylor expansion

->Taylor expansion 的 1 之後的每一項的格式:

k(x, x') = c*k1(x, x'), if c > 0 is a constant, k(x, x') is valid

-> Taylor expansion 內的每一項都是 valid kernel

(10%) Consider the optimization problem

$$minimize\ (x - 2)^2$$
$$subject\ to\ (x + 3)(x - 1) \leq 3$$

State the dual problem.

④ minimize $(x-2)^2$, subject to $(x+3)(x-1) \leq 3$

$(x+3)(x-1) - 3 \leq 0$

$\Rightarrow 3-(x+3)(x-1) \geq 0$

跟a同边

Lagrange function:

$$L(x,a) = (x-2)^2 - a[3-(x+3)(x-1)]$$

Lagrange multiplier a is non negative

$$\frac{\partial L(x,a)}{\partial x} = \frac{\partial}{\partial x}\left[x^2-4x+4+ax^2+2ax-6a\right)$$

$$= 2x-4+2ax+2a = 0$$

$$x-2+ax+a=0$$

$$x(1+a)-2+a=0$$

$$x = \frac{2-a}{1+a}$$

Eliminate x from $L(x,a)$, we get the dual representation, we "maximize":

$$\tilde{L}(a) = \left(\frac{2-a}{1+a}-2\right)^2 - a\left\{3-\left(\frac{2-a}{1+a}+3\right)\left(\frac{2-a}{1+a}-1\right)\right\}$$

subject to $a \geq 0$