NYCU Pattern Recognition, Homework 4

Deadline: May 17, 23:59

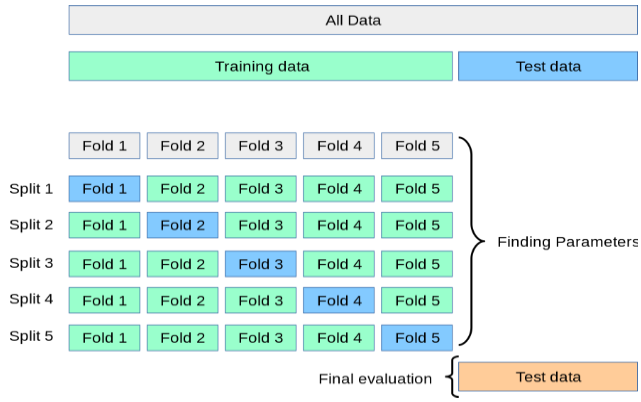
Part. 1, Coding (50%):

For this coding assignment, you are required to implement Cross-Validation and Grid Search using only NumPy. After that, you should train the SVM model from scikit-learn on the provided dataset and test the performance with the testing data. **You will get no points by simply calling** sklearn.model_selection.GridSearchCV.

(50%) K-Fold Cross-Validation & Grid Search

Requirements:

- Implement K-Fold Cross-Validation by creating a function that takes K as an argument and returns a list of K sublists.
    - Each sublist should contain two parts:
        - The first part contains the index of all training folds (index_x_train, index_y_train), for example, Fold 2 to Fold 5 in split 1.
        - The second part contains the index of the validation fold (index_x_val, index_y_val), for example, Fold 1 in split 1 .
    - You need to handle if the sample size is not divisible by K.
    - The first **n_samples % n_splits** folds should have a size of **n_samples // n_splits** + 1, and the other folds should have a size of **n_samples // n_splits**. Here, n_samples is the number of samples and n_splits is K.
    - Each of the samples should be used **exactly once** as the validation data.
    - Please **shuffle** your data before partition.



- Implement Grid Search & Cross-Validation:
    - Using sklearn.svm.SVC to train a classifier on the provided train set and perform **Grid Search** to find the best hyperparameters via cross-validation.

Criteria:

1. (10%) Implement K-fold data partitioning.

```
def cross_validation(x_train, y_train, k=5):

    # Do not modify the function name and always take 'x_train, y_train, k' as the inputs.

    # TODO HERE
    n_samples = x_train.shape[0]
    indices = np.arange(n_samples)
    np.random.shuffle(indices)
    folds = []
    '''
    let n_samples = 70, K = 8
    then first 70 % 8 = 6 folds, each has 70 // 8 + 1 = 9 samples
    other 8-6 folds, each has 70 // 8 = 8 samples
    total = 6 * 9 + 2 * 8 = 70 samples == n_samples
    '''
    size = n_samples//k + 1
    for i in range(n_samples%k):
        start = i * size
        fold = indices[start:start+size]
        folds.append(fold)

    size = n_samples // k
    for i in range(n_samples%k, k):
        start = i * size
        fold = indices[start:start+size]
        folds.append(fold)
    folds = np.asarray(folds)

    kfold = []
    for i in range(k):
        train = folds[np.arange(k)!=i]
        val = folds[i]
        kfold.append([train.ravel(), val])
    return kfold
    #return NotImplementedError
```

這個做法可以讓K不整除samples時，所有folds的數量平均，而不會有一個特別少。

2. (10%) Set the kernel parameter to 'rbf' and do grid search on the hyperparameters C and gamma to find the best values through cross-validation. Print the best hyperparameters you found. Note that we suggest using K=5 for the cross-validation.

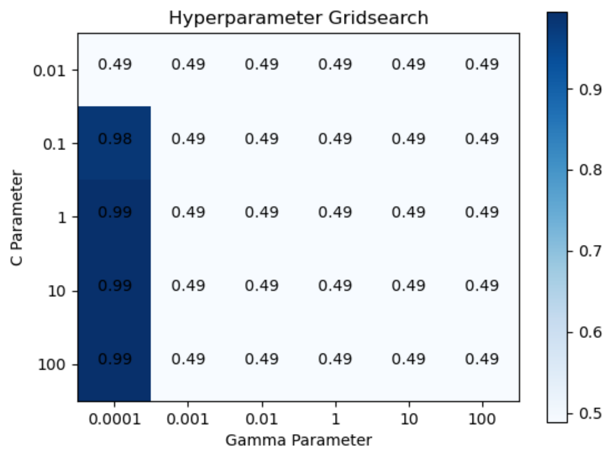
```
C = [0.01, 0.1, 1, 10, 100]
gamma = [0.0001, 0.001, 0.01, 1, 10, 100]
max_acc = 0
grid = []
kfold_data = cross_validation(x_train, y_train, 5)
```

```
print("(best_c, best_gamma) is ", best_parameters)
print(grid_array)
✓ 0.0s

(best_c, best_gamma) is  (1, 0.0001)
```

我在做了一些嘗試之後，發現當gamma足夠小時，SVC才有足夠的學習成效，最後我的best parameter是 C = 1, gamma = 0.0001

3. (10%) Plot the results of your SVM's grid search. Use "gamma" and "C" as the x and y axes, respectively, and represent the average validation score with color. Below image is just for reference.



4. (20%) Train your SVM model using the best hyperparameters found in Q2 on the entire training dataset, then evaluate its performance on the test set. Print your testing accuracy.

```
# Do Not Modify Below

best_model = SVC(C=best_parameters[0], gamma=best_parameters[1], ke
best_model.fit(x_train, y_train)

y_pred = best_model.predict(x_test)

print("Accuracy score: ", accuracy_score(y_pred, y_test))

# If your accuracy here > 0.9 then you will get full credit (20 poi
✓ 8.9s

Accuracy score:  0.995
```

Points	Testing Accuracy
20 points	acc > 0.9
10 points	0.85 <= acc <= 0.9
0 points	acc < 0.85

Part. 2, Questions (50%):

1. (10%) Show that the kernel matrix $K = [k(x_n, x_m)]_{nm}$ should be positive semidefinite is the necessary and sufficient condition for $k(x, x')$ to be a valid kernel.

kernel matrix $K = [k(x_n, x_m)]$

kernel function $k(x, x') = \phi(x)^T \phi(x')$

K is positive semidefinite $\leftrightarrow$ $k(x, x')$ is a valid kernel

$\leftarrow$: proof K is PSD

let $a = [a_1, a_2, \cdots, a_n] \subseteq R^n$

$\Rightarrow a^T K a = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, x_j)$

$\because k(x_i, x_j)$ is valid,

$\therefore k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$, $\phi(x)$ is feature map

$\Rightarrow a^T K a = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j \phi(x_i)^T \phi(x_j)$

$= \sum a_i \phi(x_i)^T \sum a_j \phi(x_j)$

$= (\sum a_i \phi(x_i))^2 \geq 0$

$\therefore K$ is positive semidefinite #

$\rightarrow$: proof $\exists \phi(x)$, $k(x, x') = \phi(x)^T \phi(x')$

let $\{x_1, x_2, \cdots, x_n\}$, $\{x_1', x_2', \cdots, x_m'\}$ be any sets of inputs

$\because K$ is PSD

$\therefore \exists b_1, b_2, \cdots b_n \in R$, $K = \sum_{i=1}^{n} \sum_{j=1}^{n} b_i b_j k(x_i, x_j)$

let $B = \begin{bmatrix} b_1 & \cdots & b_1 \\ b_2 & \cdots & b_2 \\ & \vdots & \\ b_n & \cdots & b_n \end{bmatrix}_{n \times m}$, $B' = \begin{bmatrix} b_1' & \cdots & b_1' \\ b_2' & \cdots & b_2' \\ & \vdots & \\ b_m' & \cdots & b_m' \end{bmatrix}_{m \times n}$, $C = \{C_{ij} \mid C_{ij} = k(x_i, x_j')\}$

$\Rightarrow K = BB^T$, $C = B(B')^T$

$\because K$ is PSD, $\therefore K^{\frac{1}{2}}$ is PSD

let $L = BK^{\frac{1}{2}}$, $L' = B'K^{\frac{1}{2}}$

$\Rightarrow C = L(L')^T$

$\Rightarrow k(x_i, x_j') = C_{ij} = \phi(x_i)^T \phi(x_j') = \phi(x_i) \cdot \phi(x_j)$

$\Rightarrow k(x_i, x_j')$ can be expressed as an inner product of feature maps #

2. (10%) Given a valid kernel $k_1(x, x')$, explain that $k(x, x') = exp(k_1(x, x'))$ is also a valid kernel. (Hint: Your answer may mention some terms like ____ series or ____ expansion.)

$\because k_1(x, x')$ is a valid kernel

$\Rightarrow K_1 = [k_1(x, x')] = V \Lambda V^T$,

Λ is a diagnal matrix, and diagnal values are eigenvalues of K_1

$\Rightarrow exp(k_1(x, x')) = exp(K_1) = V(exp(\Lambda))V^T = K = [k(x, x')]$

$\therefore k(x, x')$ is a valid kernel

3. (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 show its eigenvalues.

a. $k(x, x') = k_1(x, x') + x$

a. $k(x, x') = k_1(x, x') + x$

let $g(y) = y + x$,

by (6.15), $k(x, x')$ is a valid kernel

b. $k(x,x^{\prime}) = k_1(x,x^{\prime}) - 1$

b. let $k_1(x,x') = \langle x, x' \rangle$, $x = (1,3)^T$, $x' = (2,4)$

$A = [k_1(x,x)] = \begin{bmatrix} 2 & 4 \\ 6 & 12 \end{bmatrix}$

let $P_A(x) = \det(A - xI)$

$= \det\left(\begin{bmatrix} 2-x & 4 \\ 6 & 12-x \end{bmatrix}\right)$

$= (2-x)(12-x) - 24$

$= 0$

$\Rightarrow x = 0, 14$

$\Rightarrow A = P \begin{bmatrix} 0 & 0 \\ 0 & 14 \end{bmatrix} P^{-1}$, is PSD

$K = [k(x,x') - 1]$

$\Rightarrow$ eigenvalues of $K = \{-1, 13\}$

$\because -1 < 0$, $\therefore k_1(x,x') - 1$ is not a valid kernel

c. $k(x,x^{\prime}) = k_1(x,x^{\prime})^2 + exp\left(\|x\|^2\right) * exp(\|x^{\prime}\|^2)$

C. $k(x,x') = k_1(x,x')^2 + e^{\|x\|^2} * e^{\|x'\|^2}$

let $K = [k(x,x')]$, $\forall \vec{a}$

$a^T K a = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_i, x_j')$

$= \sum \sum \underbrace{a_i a_j}_{\geq 0} [\underbrace{k_1(x_i, x_j)^2}_{\geq 0} + \underbrace{e^{x_i^2}}_{>0} + \underbrace{e^{(x_j')^2}}_{>0}]$

$\therefore k(x,x')$ is a valid kernel

d. $k(x,x^{\prime}) = k_1(x,x^{\prime})^2 + exp\left(k_1(x,x^{\prime})\right) - 1$

D. let $K = [k(x,x')]$, $\forall \vec{a}$

$a^T K a = \sum \sum a_i a_j [k_1(x_i, x_j)^2 + \exp(k_1(x_i, x_j)) - 1]$

$\because k_1(x_i, x_j) \geq 0$, $\therefore \exp(k_1(x_i, x_j)) \geq 1$

$\Rightarrow a^T K a \geq 0$

$\therefore K$ is valid kernel

4. Consider the optimization problem

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

State the dual problem. (Full points by completing the following equations)

$L(x, \lambda)$ = _ (x - 2)^2 + λ [(x+4)(x-1) - 3]_

$\nabla_x L(x, \lambda)$ = ___ 2(x - 2) + λ (2x + 3)_____

when $\nabla_x L(x, \lambda) = 0$,

x = ____ x = (4 - 3 λ)/2_____

$L(x, \lambda) = L(\lambda)$ = _____ (4 - 3 λ)^2/4 + λ [(8 - 3 λ)(2 - 3 λ) - 3]_______