NCTU Pattern Recognition, Homework 4

Deadline: May 25, 23:59

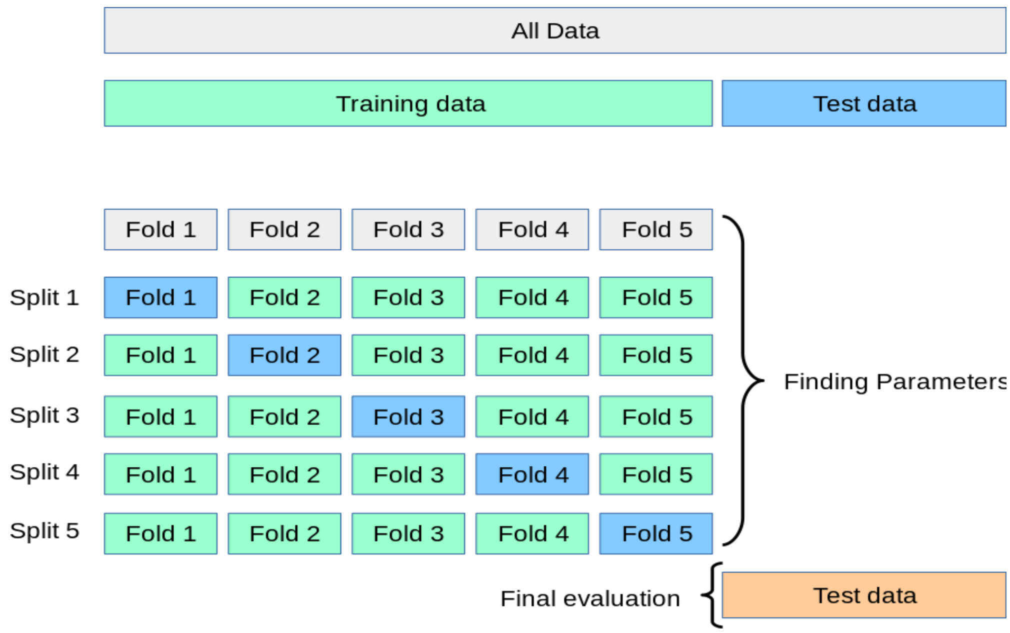
Part. 1, Coding (50%):

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)

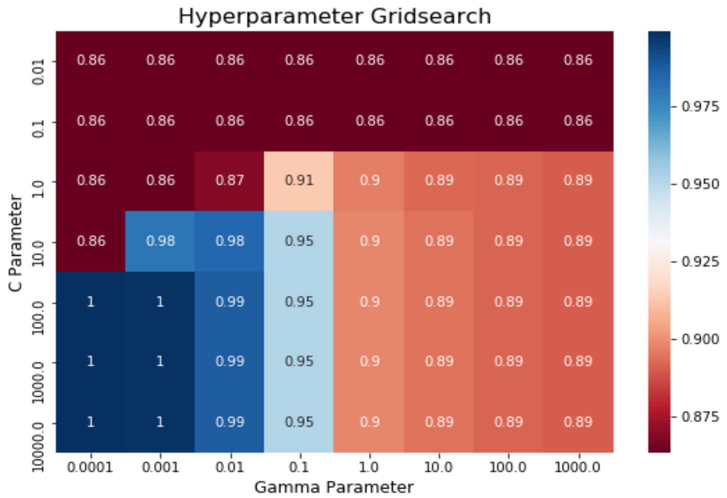
   Note: You need to handle if the sample size is not divisible by K. Using the strategy from sklearn. The first n_samples % n_splits folds have size n_samples // n_splits + 1, other folds have size n_samples // n_splits, where n_samples is the number of samples, n_splits is K, % stands for modulus, //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

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: This image is for reference, not the answer
   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.

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

[sol]:

Q1:

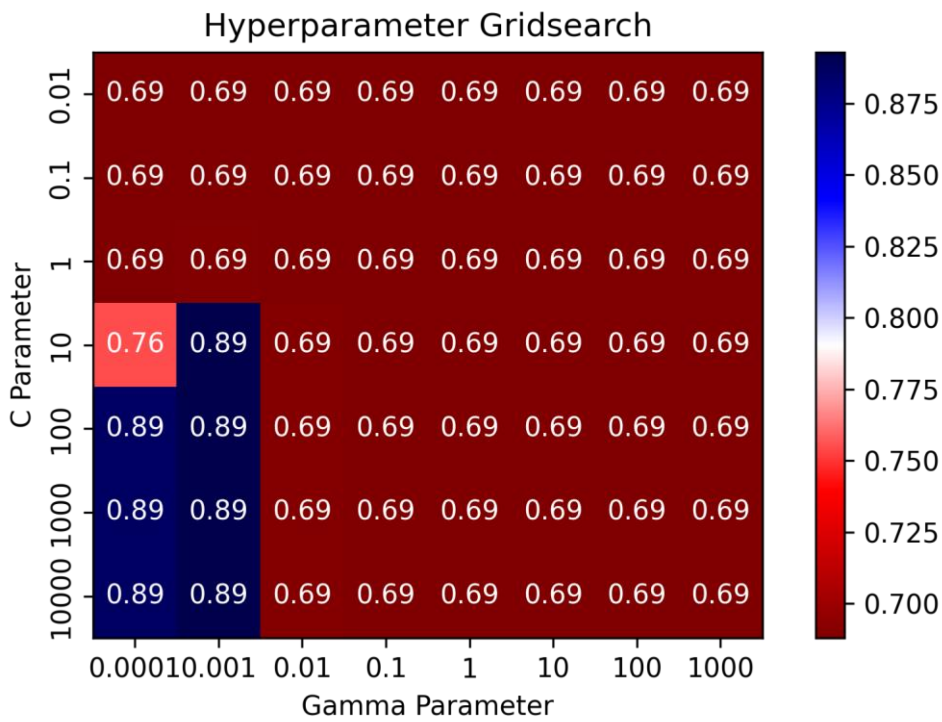
To write the cross_validation library, I write it by first finding out how many samples ther e are in x_train, creating indices and randomly disordering them, and then taking samples f rom them. The samples are then stored in a list format according to the topic settings.

Q2:

There are two important parameters in the svm function, c and gamma, which represent the penalty coef ficient and the quoted value in the kernel function. c is mainly used to penalize the error value, the high er the value, the smaller the error will be, but the disadvantage is that it is easy to overfit, the smaller th e value, the larger the error value, the worse the generalization ability. These two parameters have a co mplementary relationship, so the suitable parameter is found by grid search. In this problem, we first set the range of parameters to be explored, and use two for loops for grid search, and use k-fold dataset for svm model training, and finally use the average accuracy as the index to evaluate each set of hyperpara meters

Q3:

Using heat map to plot the grid search results of SVM, x, y represent the hyperparameters of gamma and c respectively, red represents low average accuracy, blue represents high ac curacy, from the results it is found that the hyperparameters in the lower left corner have hi gh accuracy, if you want to improve the accuracy, you can target the hyperparameter grid s earch in the lower left corner for a more detailed local segmentation.



Q4:

According to the heat map in question 3, I reduced the range of c and gamma respectively, and adjusted the number of k-fold to 55, which could improve the accuracy to 0.90625.

Question 4

Train your SVM model by the best parameters you found from question 2 on the whole training set and evaluate the performance on the test set.

In [208…

```
kfold_data = cross_validation(x_train, y_train, k=55)
cand_C = [10, 10.5, 11 ,12 ,13 ,14,15,16,17,18,19,20]
cand_gamma = [1e-3, 1.2*1e-3, 1.4*1e-3, 1.6*1e-3,1.8*1e-3]
gridsearch, best_parameters = svm_gridsearch(x_train, y_train, kfold_data, cand_C, cand_gamma)
print(f'Best parameter (C, gamma): {best_parameters}')
```

```
C=10, gamma=0.001, avg acc=0.90
C=10, gamma=0.0012, avg acc=0.90
C=10, gamma=0.0014, avg acc=0.90
C=10, gamma=0.0016, avg acc=0.90
C=10, gamma=0.0018000000000000002, avg acc=0.89
C=10.5, gamma=0.001, avg acc=0.90
C=10.5, gamma=0.0012, avg acc=0.90
C=10.5, gamma=0.0014, avg acc=0.90
C=10.5, gamma=0.0016, avg acc=0.90
C=10.5, gamma=0.0018000000000000002, avg acc=0.89
C=11, gamma=0.001, avg acc=0.90
C=11, gamma=0.0012, avg acc=0.90
C=11, gamma=0.0014, avg acc=0.90
C=11, gamma=0.0016, avg acc=0.90
C=11, gamma=0.0018000000000000002, avg acc=0.89
C=12, gamma=0.001, avg acc=0.90
C=12, gamma=0.0012, avg acc=0.90
C=12, gamma=0.0014, avg acc=0.90
C=12, gamma=0.0016, avg acc=0.90
C=12, gamma=0.0018000000000000002, avg acc=0.89
C=13, gamma=0.001, avg acc=0.90
C=13, gamma=0.0012, avg acc=0.90
C=13, gamma=0.0014, avg acc=0.90
C=13, gamma=0.0016, avg acc=0.90
C=13, gamma=0.0018000000000000002, avg acc=0.89
C=14, gamma=0.001, avg acc=0.90
C=14, gamma=0.0012, avg acc=0.90
C=14, gamma=0.0014, avg acc=0.90
C=14, gamma=0.0016, avg acc=0.90
C=14, gamma=0.0018000000000000002, avg acc=0.89
C=15, gamma=0.001, avg acc=0.90
C=15, gamma=0.0012, avg acc=0.90
C=15, gamma=0.0014, avg acc=0.90
C=15, gamma=0.0016, avg acc=0.90
C=15, gamma=0.0018000000000000002, avg acc=0.89
C=16, gamma=0.001, avg acc=0.90
C=16, gamma=0.0012, avg acc=0.90
C=16, gamma=0.0014, avg acc=0.90
C=16, gamma=0.0016, avg acc=0.90
C=16, gamma=0.0018000000000000002, avg acc=0.89
C=17, gamma=0.001, avg acc=0.90
C=17, gamma=0.0012, avg acc=0.90
C=17, gamma=0.0014, avg acc=0.90
C=17, gamma=0.0016, avg acc=0.90
C=17, gamma=0.0018000000000000002, avg acc=0.89
C=18, gamma=0.001, avg acc=0.90
C=18, gamma=0.0012, avg acc=0.90
C=18, gamma=0.0014, avg acc=0.90
C=18, gamma=0.0016, avg acc=0.90
C=18, gamma=0.0018000000000000002, avg acc=0.89
C=19, gamma=0.001, avg acc=0.90
C=19, gamma=0.0012, avg acc=0.90
C=19, gamma=0.0014, avg acc=0.90
C=19, gamma=0.0016, avg acc=0.90
C=19, gamma=0.0018000000000000002, avg acc=0.89
C=20, gamma=0.001, avg acc=0.90
C=20, gamma=0.0012, avg acc=0.90
C=20, gamma=0.0014, avg acc=0.90
C=20, gamma=0.0016, avg acc=0.90
C=20, gamma=0.0018000000000000002, avg acc=0.89
Best parameter (C, gamma): (20, 0.0014, 0.9018181818181813)
```

In [209…

```
best_C, best_gamma, _ = best_parameters
best_model = SVC(C=best_C, kernel='rbf', gamma=best_gamma)
best_model.fit(x_train, y_train)
y_pred = best_model.predict(x_test)
print("Accuracy score: ", accuracy_score(y_pred, y_test))
```

```
Accuracy score:  0.90625
```

Part. 2, Questions (50%):

1. (10%) Given a valid kernel $k_1(x, x')$, prove that the following proposed functions are or are not valid kernels.
   a. $k(x, x') = (k_1(x, x'))^2 + (k_1(x, x') + 1)^2$
   b. $k(x, x') = (k_1(x, x'))^2 + \exp(||x||^2) * \exp(||x'||^2)$

2. (10%) Show that the kernel matrix s $\mathbf{K} = [k(\mathbf{x}_n, \mathbf{x}_m)]_{nm}$ hould be positive semidefinite is the necessary and sufficient condition for to be $k(\mathbf{x}, \mathbf{x}')$ a valid kernel.

3. (10%) Consider the dual formulation of the least-squares linear regression problem given on page 6 in the ppt of Kernel Methods. Show that the solution for the components $\mathbf{a}_n$ of the vector $\mathbf{a}$ can be expressed as a linear combination of the elements of the vector $\varphi(\mathbf{x}_n)$. Denoting these coefficients by the vector $\mathbf{w}$, show that the dual of the dual formulation is given by the original representation in terms of the parameter vector $\mathbf{w}$.

4. (10%) Prove that the Gaussian kernel defined by (eq 1) is valid and show the function φ (x), where $x \in \mathbf{R}^1$.
   $$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2\right) = \phi(x)^{\mathrm{T}}\phi(x')$$
   (eq1)

5. (10%) Consider the optimization problem

   minimize $(x - 2)^2$

   subject to $(x+3)(x-1) \leqq 2$

   State the dual problem.

學號： 410557021
姓名： 江衍涵

Part 2:

1.

(a) $k(x, x') = \left(k_1(x, x')\right)^2 + \left(k_1(x, x') + 1\right)^2$

$\Rightarrow (x^T x')^2 + (x^T x' + 1)^2$

$\Rightarrow \left(\sum_{i=1}^{d} x_i x_i'\right)^2 + \left(\sum_{i=1}^{d} x_i x_i' + 1\right)^2$

$\Rightarrow \sum_i \sum_j x_i x_j x_i' x_j' + 1 + 2 \sum_i x_i x_i' + \sum_i \sum_j x_i x_j x_i' x_j'$

$\Rightarrow 1 + 2 \sum_i x_i x_i' + 2 \sum_i \sum_j (x_i x_j)(x_i' x_j')$

$\Rightarrow 1 + \sum_i (\sqrt{2} x_i)(\sqrt{2} x_i') + \sum_i \sum_j (\sqrt{2} x_i x_j)(\sqrt{2} x_i' x_j')$

Thus $k(x, x') = \phi(x)^T \phi(x')$ with

$\phi(x) = [1, \sqrt{2} x_1, \ldots, \sqrt{2} x_d, \sqrt{2} x_1 x_1, \sqrt{2} x_1 x_2, \ldots \sqrt{2} x_1 x_d, \sqrt{2} x_2 x_1, \ldots \sqrt{2} x_d x_d]^T$

It's a valid kernel.

1.

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

$$k_1(x_1, x')^2 = (x^T x')^2 \Rightarrow (x_1 z_1 + x_2 z_2)^2$$
$$\Rightarrow x_1^2 z_1^2 + 2 x_1 z_1 x_2 z_2 + x_2^2 z_2^2$$
$$\Rightarrow (x_1^2, \sqrt{2} x_1 x_2, x_2^2)(z_1^2, \sqrt{2} z_1 z_2, z_2^2)$$
$$\Rightarrow \phi(x)^T \phi(z)$$

Let $\psi : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ be a symmetric positive definite kernel.

Let $f : \mathbb{R}^N \to \mathbb{R}^N$ be any function.

Then $k(x, x') = f(x) \psi(x, x') f(x')$ is a symmetric positive definite kernel.

The symmetry is immediate because ψ is symmetric and multiplication is commutative.

Let $x_1, \dots, x_M \in \mathbb{R}^N$ and $c_1, \dots, c_M \in \mathbb{R}^N$, Then

$$\sum_{i=1}^{M} \sum_{j=1}^{M} c_i k(x, x') c_j = \sum_{i=1}^{M} \sum_{j=1}^{M} c_i f(x_i) \psi(x_i, x_j) f(x_j c_j)$$
$$= \sum_{i=1}^{M} \sum_{j=1}^{M} (c_i f(x_i)) \psi(x_i, x_j) (f(x_j) c_j)$$

Let $d_i = c_i f(x_i)$.

Then, since ψ is positive definite.

$$\sum_{i=1}^{M} \sum_{j=1}^{M} c_i k(x_i, x_j) c_j = \sum_{i=1}^{M} \sum_{j=1}^{M} d_i \psi(x_i, x_j) d_j \geq 0$$

2.

If we consider the Gram matrix, K, corresponding to the l.h.S. of PRML eq 6.19, we have

$$(K)_{ij} = k(x_i, x_j) = k_3(\phi(x_i), \phi(x_j)) = (K_3)_{ij}$$

where K_3 is the Gram matrix corresponding to $k_3(\cdot,\cdot)$. Since $k_3(\cdot,\cdot)$ is a valid kernel,

$$u^T K u = u^T K_3 u \geq 0$$

For PRML (eq 6.20), let $K = X^T A X$,

so that $(K)_{ij} = x_i^T A x_j$, and consider

$$u^T K u = u^T X^T A X u$$
$$= v^T A v \geq 0$$

where, $v = Xu$ and we have used that

A is positive semidefinite

3. kernel function is given by the relation:

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

a_n can be written as:

$$a_n = -\frac{1}{\lambda}\{w^T \phi(x_n) - t_n\}$$

We can derive:

$$a_n = -\frac{1}{\lambda}\{w^T \phi(x_n) - t_n\}$$

$$= -\frac{1}{\lambda}\{w_1 \phi_1(x_n) + w_2 \phi_2(x_n) + \dots w_m \phi_m(x_n) - t_n\}$$

$$= -\frac{w_1}{\lambda}\phi_1(x_n) - \frac{w_2}{\lambda}\phi_2(x_n) - \dots \frac{w_m}{\lambda}\phi_m(x_n) + \underline{\underline{\frac{t_n}{\lambda}}}$$

$$= \left(c_n - \frac{w_1}{\lambda}\right)\phi_1(x_n) + \left(c_n - \frac{w_2}{\lambda}\right)\phi_2(x_n) + \dots + \left(c_n - \frac{w_m}{\lambda}\right)\phi_m(x_n)$$

We have defined

$$c_n = \frac{t_n/\lambda}{\phi_1(x_n) + \phi_2(x_n) + \dots \phi_m(x_n)}$$

We observe from the above differential equation, we can see that a_n is a linear combination of $\phi(x_n)$.

First, we substitute $K = \phi\phi^T$ into:

$$J(a) = \frac{1}{2} a^T K K a - a^T K t + \frac{1}{2} t^T t + \frac{\lambda}{2} a^T K a$$

$$\Rightarrow J(a) = \frac{1}{2} a^T \phi\phi^T \phi\phi^T a - a^T \phi\phi^T t + \frac{1}{2} t^T t + \frac{\lambda}{2} a^T \phi\phi^T a \quad \text{— (1)}$$

$$w = -\frac{1}{\lambda} \sum_{n=1}^{N} \{ w^T \phi(x_n) - t_n \} \phi(x_n) = \sum_{n=1}^{N} a_n \phi(x_n) = \phi^T a \quad \text{— (2)}$$

Next, we substitue eq (2) into eq (1)

$$J(a) = \frac{1}{2} \underbrace{a^T \phi}_{w^T} \phi^T \underbrace{\phi\phi^T a}_{w} - \underbrace{a^T \phi}_{w^T} \phi^T t + \frac{1}{2} t^T t + \frac{\lambda}{2} \underbrace{a^T \phi}_{w^T} \underbrace{\phi^T a}_{w}$$

We have proof:

$$J(w) = \frac{1}{2} \sum_{n=1}^{N} \{ w^T \phi(x_n) - t_n \}^2 + \frac{\lambda}{2} w^T w$$

4.

$$k(x,x') = \exp\left(\frac{-\|x-x'\|^2}{\sigma^2}\right) = \exp\left(\frac{-\|x\|^2 - \|x'\|^2 + 2x^T x'}{\sigma^2}\right)$$

$$= \left(\exp\left(\frac{-\|x\|}{\sigma^2}\right)\exp\left(\frac{-\|x'\|}{\sigma^2}\right)\right)\exp\left(\frac{2x^T \cdot x'}{\sigma^2}\right)$$

$$= g(x)\cdot g(x')\cdot \exp(k_1(x,x'))$$

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

where $g(x)\,g(x')$ is a kernel according PRML eq 6.18, and

$\exp(k_1(x,x'))$ is a kernel according PRML eq.6.16.

And $x^T \cdot x'$ is a valid linear, σ^2 is postive.

The product of kernel is a valid kernel.

5. minimize $(x-2)^2$

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

dual function : $g(\lambda) = \inf_x \left(x^T (A + \lambda I) x + 2b^T x - \lambda \right)$