NCTU Introduction to Machine Learning, Homework 4

黃品云 109550017

Part. 1, Coding (50%):

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)

```
# test the K-fold cross-validation function with example below
X = np.arange(20)
kf = cross_validation(X, X, k=5)
for i, (train_index, val_index) in enumerate(kf):
    print("Split: %s, Training index: %s, Validation index: %s" % (i+1, train_index, val_index))
```

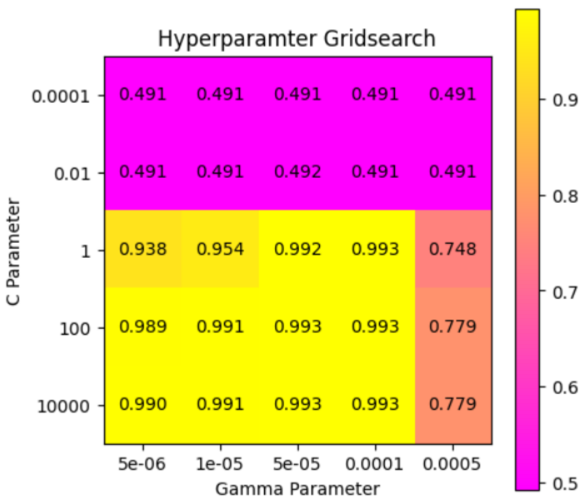
```
Split: 1, Training index: [ 4 19  8  2 15  6 17 13 18 11  1  9 10  3  0 14], Validation index: [12 16  5  7]
Split: 2, Training index: [12 16  5  7 15  6 17 13 18 11  1  9 10  3  0 14], Validation index: [ 4 19  8  2]
Split: 3, Training index: [12 16  5  7  4 19  8  2 18 11  1  9 10  3  0 14], Validation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

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(f'Best C={best_hyperparameter[0]}, gamma={best_hyperparameter[1]}')
```

```
Best C=1, 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.



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.

NO y_test provided

```python
import pickle
```

```python
best_model = SVC(C=1, kernel='rbf', gamma=0.0001)
best_model.fit(x_train, y_train)
```

```
SVC
SVC(C=1, gamma=0.0001)
```

```python
y_pred = best_model.predict(x_test)
```

```python
with open('model.pickle', 'wb') as pkl_file:
    pickle.dump(best_model, pkl_file, protocol=pickle.HIGHEST_PROTOCOL)
```

```python
np.save('y_pred.npy', y_pred)
```

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.

The kernel function $k(x_n, x_m)$ 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. K is positive semidefinite, so all eigenvalues are non-negative.

Define a feature mapping into a n-dimensional space where the t-th bit in feature expansion for example x_i is $\phi_t(x_i) = \sqrt{\lambda_t}\, V_{ti}$

The inner product is

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

(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.

$$exp(k_1(x,x')) = exp(0) + exp(0)\,k_1(x,x') + \frac{exp(0)}{2!} \cdot k_1(x,x')^2 + \frac{exp(0)}{3!} \cdot k_1(x,x')^3 + \cdots$$

$$\Rightarrow exp(k_1(x,x')) = 1 + k_1(x,x') + \frac{1}{2} k_1(x,x')^2 + \frac{1}{6} k_1(x,x')^3 + \cdots$$

according to $k(x,x') = c k_1(x,x')$ (6.13) and

$k(x,x') = k_1(x,x') + k_2(x,x')$ (6.17)

$k(x,x') = k_1(x,x')\,k_2(x,x')$ (6.18)

and $k_1(x,x')$ is a valid kernel, we can derive that the exponential of a valid kernel is also a 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 show its eigenvalues.

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

according to $k(x,x') = k_1(x,x') + k_2(x,x')$ (6.17)
and any positive constant is a valid kernel
$\Rightarrow$ $\underline{k(x,x') = k_1(x,x') + 1 \text{ is a valid kernel}}$, since $k_1(x,x')$ and 1 are valid kernels. #

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

according to $k(x,x') = ck_1(x,x')$ (6.13) where $c > 0$ is a constant, "-1" is not a valid kernel cause we can't find -1 with $c > 0$.
Therefore, $\underline{k(x,x') = k_1(x,x') - 1 \text{ is not a valid kernel.}}$ #

e.g.

assume $k_1(x,x')$ is 1 when $x = x'$ and 0 otherwise. G is positive and semidefinite since it's an identity matrix. If $k(x,x') = k_1(x,x') - 1$, then $k(x,x')$ is 0 when $x = x'$ and -1 otherwise. G' is

$$\begin{bmatrix} 0 & -1 & \cdots & -1 \\ -1 & 0 & & \vdots \\ \vdots & & \ddots & \vdots \\ -1 & \cdots & & 0 \end{bmatrix}$$

at this time, and it is not positive and semidefinite.

$$\det(G - \lambda I) = -\lambda^3 - 2 = 0 \Rightarrow \lambda = -\sqrt[3]{2}, \frac{\sqrt[3]{2} \pm \sqrt{2^{\frac{2}{3}} - 4 \cdot 2^{\frac{2}{3}}}}{2}$$

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

(1) according to $k(x,x') = k_1(x,x')\, k_2(x,x')$ (6.18)
$k(x,x')$ is a valid kernel when $k_1(x,x')$ and
$k_2(x,x')$ are valid kernels, so we can derive
that $k_1(x,x')^2$ is a valid kernel.

(2) according to $k(x,x') = f(x)\, k_1(x,x')\, f(x')$ (6.14)
and any positive constant is a valid kernel, we
can assume

$$\exp(\|x\|^2) = f(x)$$
$$1 = k_1(x,x')$$
$$\exp(\|x'\|^2) = f(x')$$

Thus, $\exp(\|x\|^2) \times \exp(\|x'\|^2)$ is a valid kernel #

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

$$k(x,x') = k_1(x,x')^2 + \left[1 + k_1(x,x') + \tfrac{1}{2}k_1(x,x')^2 + \dots\right] - 1$$

$$= k_1(x,x')^2 + k_1(x,x') + \tfrac{1}{2}k_1(x,x')^2 + \dots$$

according to $k(x,x') = c\,k_1(x,x')$ (6.13) and
$k(x,x') = k_1(x,x') + k_2(x,x')$ (6.17)
$k(x,x') = k_1(x,x')\, k_2(x,x')$ (6.18)

$k_1(x,x')^2 + \exp(k_1(x,x')) - 1$ is a valid kernel. #

(10%) Consider the optimization problem

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

State the dual problem.

We can use Lagrangian function to solve the optimization problem

$L(x, \lambda) = f(x) + \lambda g(x)$ and $g(x)$ is constant.

We assume $g(x) = (x+3)(x-1) \leq 3$

$\Rightarrow g(x) = x^2 + 2x - 6 \leq 0$

and $f(x) = (x-2)^2$

We can get $L(x,\lambda) = (x-2)^2 + \lambda(x^2+2x-6)$

$= (1+\lambda)x^2 + (-4+2\lambda)x + (4-6\lambda)$

Then, we use gradient to find substitute for x with λ

$\frac{\partial L(x,\lambda)}{x} = 2(1+\lambda)x - 4 + 2\lambda = 0 \Rightarrow x = \frac{4-2\lambda}{2(1+\lambda)} = \frac{2-\lambda}{1+\lambda}$

$\frac{\partial L(x,\lambda)}{\lambda} = x^2 + 2x - 6 = 0$ (original constraint)

Now, we can rewrite $L(x, \lambda)$ with substitute λ for x, and we get $L(\lambda) = \frac{-\lambda^2+4\lambda-4}{1+\lambda} + (4-6\lambda)$, which is Lagranian dual function.

Since we want to minimize the original problem, we can perform the same behavior by maximize the corresponding dual problem

The dual problem: maximize $\frac{-\lambda^2+4\lambda-4}{1+\lambda} + (4-6\lambda)$

subject to $\lambda \geq 0$