NYCU Pattern Recognition, Homework 4

311551059, 陳昱丞

Part. 1, Coding (50%):

You should type the answer and also screenshot at the same time. Otherwise, no points will be given. The screenshot and the figures we provided below are just examples. **The results below are not guaranteed to be correct.** Please convert it to a pdf file before submission. You should use English to answer the questions. After reading this paragraph, you can delete it.

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
    indices = np.arange(len(x_train))
    np.random.seed(0)
    np.random.shuffle(indices)
    folds_idx = np.array_split(indices, k)  # Allows indices to be an integer that does not equally divide the axis.
    folds_idx = np.array(folds_idx)

    k_fold_data = []
    for i in range(k):
        train_idx = [j for j in range(k) if i != j]
        k_fold_data.append([np.concatenate((folds_idx[train_idx]), axis=None), folds_idx[i]])

    return k_fold_data
```

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.

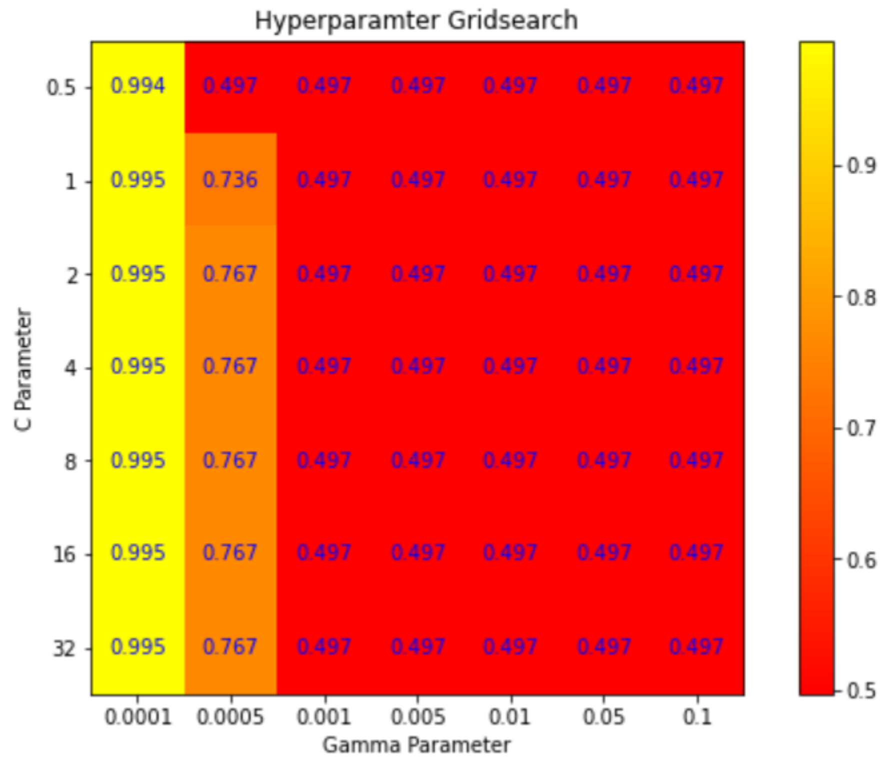
(best_c, best_gamma) is (1, 0.0001).

```
best_c, best_gamma = None, None

# TODO HERE
# k-Flod Cross Validation and Grid Search
kfold_data = cross_validation(x_train, y_train, k=5)
c_grid = [0.5, 1, 2, 4, 8, 16, 32]
gamma_grid = [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1]
best_score = 0
grid_score = []
for i in range(len(c_grid)):
    li = []
    for j in range(len(gamma_grid)):
        clf = SVC(C=c_grid[i], gamma=gamma_grid[j], kernel='rbf')
        score = 0
        for train_idx, val_idx in kfold_data:
            clf.fit(x_train[train_idx], y_train[train_idx])
            y_pred = clf.predict(x_train[val_idx])
            score += accuracy_score(y_pred, y_train[val_idx])

        avg_score = score / len(kfold_data)
        li.append(avg_score)
        print(f'C={c_grid[i]}, gamma={gamma_grid[j]}, score = {avg_score}')
        if avg_score > best_score:
            best_score = avg_score
            best_c = c_grid[i]
            best_gamma = gamma_grid[j]

    grid_score.append(li)

best_parameters=(best_c, best_gamma)
```

```
print("(best_c, best_gamma) is ", best_parameters)
```

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



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.

Accuracy: 0.995

```
# Do Not Modify Below

best_model = SVC(C=best_parameters[0], gamma=best_parameters[1], kernel='rbf')
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 points).
```

```
Accuracy score:  0.995
```

Part. 2, Questions (50%):

(10%) 1. 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.

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: ϕ: $x_{i} \longrightarrow \left(\sqrt{\lambda_{t}}\, v_{ti}\right)_{t=1}^{n} \in R^{n}$, we find that:

$$\phi\left(x_{i}\right)^{T}\phi\left(x_{j}\right) = \sum_{t=1}^{n} \lambda_{t} v_{ti} v_{tj} = \left(V\Lambda V^{T}\right)_{ij} = K_{ij} = k(x_{i}, x_{j})$$

(10%) 2. Given a valid kernel $k_{1}(x, x')$, explain that $k\left(x, x'\right) = exp(k_{1}(x, x'))$ is also a valid kernel. (Hint: Your answer may mention some terms like ____ series or ____expansion.)

We know that the Taylor expansion of e^{x} is $1 + \frac{x}{1!} + \frac{x^{2}}{2!} + \frac{x^{3}}{3!} + ...$

Let $K = k_{1}(x, x')$

So $exp\ (K) = 1 + K + \frac{1}{2}K^{2} + \frac{1}{6}K^{3} + ...$

This is a polynomial with nonnegative coefficients. By (6.15), $exp\ (K)$ is a valid kernel.

(20%) 3. 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. $k\left(x, x'\right) = k_{1}\left(x, x'\right) + x$

b. $k\left(x, x'\right) = k_{1}\left(x, x'\right) - 1$

c. $k\left(x, x'\right) = k_{1}\left(x, x'\right)^{2} + exp\left(||x||^{2}\right) * exp\left(\left|\left|x'\right|\right|^{2}\right)$

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

3. (a) valid.

(b) not valid.

Let $k_1(x,x') = x^T x'$,

then $k(x,x') = k_1(x,x') - 1 = x^T x' - 1$

Suppose we get two 1-D input: $x_1 = 1$, $x_2 = 2$

By definition, $K = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) \\ k(x_2,x_1) & k(x_2,x_2) \end{bmatrix}$

$= \begin{bmatrix} x_1^2 - 1 & x_1 x_2 - 1 \\ x_2 x_1 - 1 & x_2^2 - 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix}$

The eigenvalues of K are $\lambda = \frac{3 \pm \sqrt{13}}{2}$,

where $\frac{3-\sqrt{13}}{2} < 0$, so $k(x,x')$ is not valid.

(c) valid.

Because exponential function is always positive,

so $k_1(x,x')^2 + exp(\|x\|^2)\, exp(\|x'\|^2)$ can be viewed as

$k_1(x,x')^2 + T$ where T is a positive constant. Then

$k(x,x') = k_1(x,x')^2 + T$ is a polynomial with non-negative

coefficients, satisfying (6.15). So $k(x,x')$

is valid.

(d) not valid.

From (6.15), we know that $k_1(x,x')^2$ is valid.

From (6.16), we know that $exp(k_1(x,x'))$ is valid.

From (6.17), we know that $k_1(x,x')^2 + exp(k_1(x,x'))$ is valid.

So we can let another valid kernel

$k_2(x,x') = k_1(x,x')^2 + exp(k_1(x,x'))$, then

$k(x,x') = k_2(x,x') - 1$. And by (b), we know

that it is not valid.

(10%) 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)

4.

$$L(x,\lambda) = \underline{(x-2)^2 - \lambda(x^2+3x-7)} ※$$

$$\nabla_x L(x,\lambda) = \underline{2x-4 - 2\lambda x - 3\lambda} ※$$

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

$$\Rightarrow 2x-4-2\lambda x-3\lambda = 0$$

$$\Rightarrow (2-2\lambda)x = 3\lambda+4$$

$$\Rightarrow \underline{x = \frac{4+3\lambda}{2-2\lambda}} ※$$

$$L(x,\lambda) = L(\lambda) = \left(\frac{4+3\lambda}{2-2\lambda} - 2\right)^2 - \lambda\left(\left(\frac{4+3\lambda}{2-2\lambda}\right)^2 + 3\left(\frac{4+3\lambda}{2-2\lambda}\right) - 7\right)$$

$$= \underline{\frac{\lambda(\lambda-1)(37\lambda+12)}{4(\lambda-1)^2}} ※$$