**Machine Learning Homework 5 Report**

**0756079 資科工碩一 陳冠聞**

Linear kernel is defined as (u'\*v)

Polynomial kernel is defined as ((gamma\*u'\*v + coef0)^degree)

RBF kernel is defined as (exp(-gamma\*|u-v|^2))

Linear + RBF is defined as lam\_1\*(u'\*v) + lam\_2\*(exp(-gamma\*|u-v|^2))

1. **Use different kernel functions**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Linear | Polynomial | RBF |
| Accuracy | 95.08% | 34.68% | 95.32% |

All use default parameters, i.e.

For Linear kernel, we use C=1.

For Polynomial kernel, we use C = 1, gamma = 1/784, degree = 3, and coef0 = 0

For RBF kernel, we use C = 1, gamma = 1/784 for RBF kernel

We can see that both Linear Kernel & RBF kernel perform good, but polynomial perform bad, it’s because the default degree 3, gamma about 1e-3 and coef=0 is not suit for this dataset, if we lower the degree, or set coef=1, or set gamma=1e-2 will all make polynomial kernel reach 90% accuracy.

# Code as follow



In [1], import necessary package

In [2], read data which have been preprocessed to fit libsvm format.

In [3], Train SVM model with different kernels and test them.

1. **Grid search**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Linear | Polynomial | RBF |
| Accuracy | 95%(#1) | 97.68% (#2) | 98.08% |

All use best parameters found in grid search for 5-fold cross validation

For Linear kernel, we use C = 2.0, which perform best for c in [1, 2, 3, 4, 5]

For Polynomial kernel, we use C =2.0, gamma=0.1, degree = 2, and coef = 0 , which perform best for c in [1, 2, 3], gamma g in [1e-3, 1e-2, 1e-1], degree d in [3,2,1]

Coef r in [0,1]

For RBF kernel, we use C = 3.0, gamma = 0.01 which perform best for c in

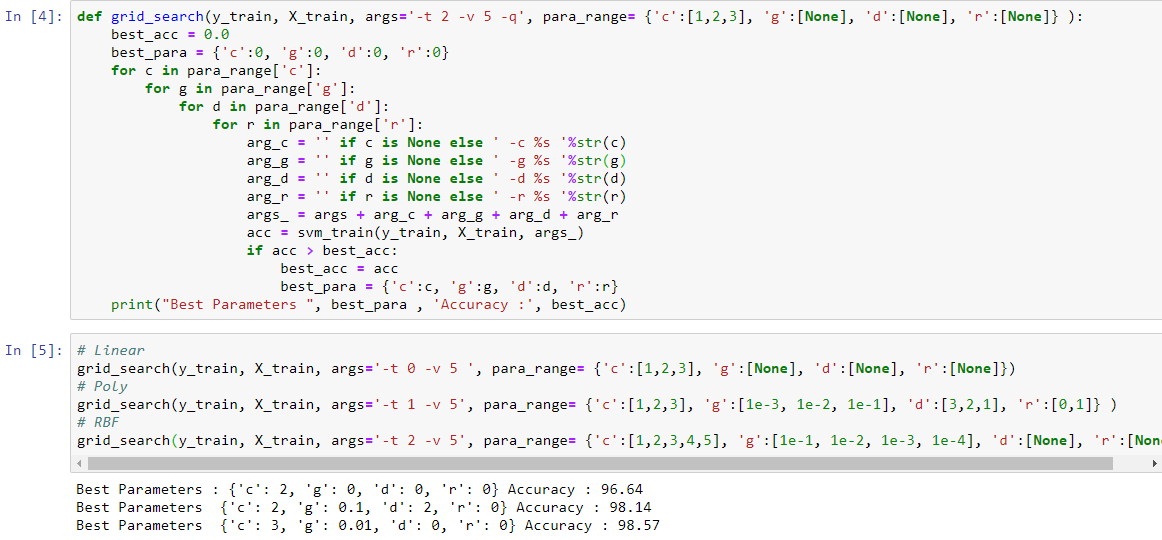
[1, 2, 3, 4, 5], gamma g in [1e-1, 1e-2, 1e-3, 1e-4, 1e-5]

(#1) We can see that the accuracy of linear kernel is lower than default setting, it’s because we use parameters that perform best in 5-fold but it’s possible it performs worse in real test data.

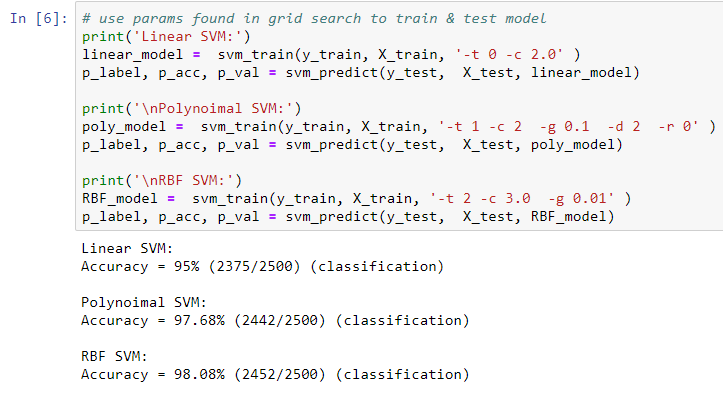
(#2) The accuracy of Polynomial Kernel improves quite a lot, but almost all setting can reach 90% above accuracy except default setting.

The accuracy of RBF kernel is improved by 2.76%.

Code as follow



In [4], [5] we use grid search to find best parameters for each model



In [6] , we apply parameters found in grid search to train model and test

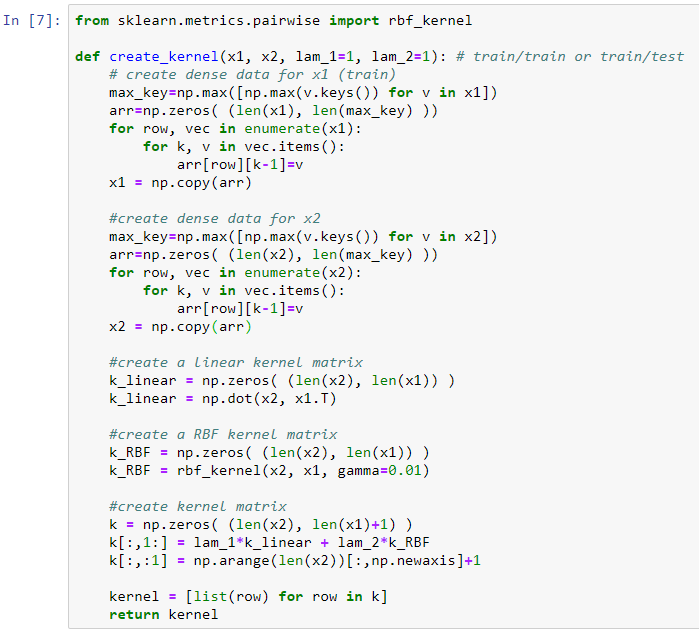
1. **linear kernel + RBF kernel**

|  |  |  |
| --- | --- | --- |
|  | RBF | Linear + RBF |
| Accuracy | 98.08% | 96% |

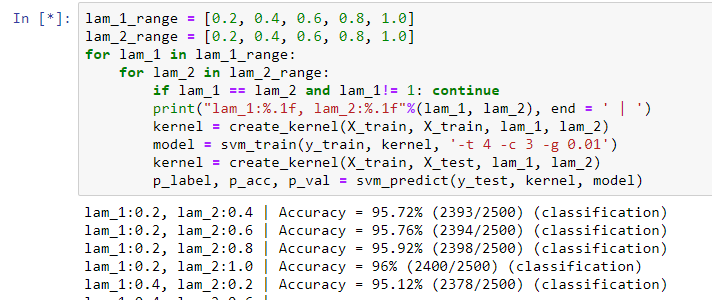
We do experiment on Linear + RBF kernel by fix c= 3.0 g=0.02 to compare with RBF kernel in question 2 and only adjust lam\_1 and lam\_2 for both lam\_1 and lam\_2 are in [0.2, 0.4, 0.6, 0.8, 1.0].

Result show that best accuracy for Linear + RBF is 96% for lam\_1=0.2 and lam\_2=1.0 in our setting is outperformed by pure RBF kernel, indicate that RBF kernel might be the most appropriate kernel for this dataset.

Codes are in next page



In [7] , we define the function to compute the kernel



In [8] , we use different weight for linear & RBF to compute kernel, and train & test model according to the computed kernel.

For more code detail, you can visit <https://github.com/aa10402tw/Machine-Learning-Implementation/tree/master/HW5/libsvm-3.23/python>