CS6220-01 Data Mining Techniques – Fall 2016

Assignment 2

Qian Cao

**Part 1 - Decision Tree**

**Before pruning:**

Top 7,4,0.95

Education gain = 0.149761440768

1.(High School) 5, 1, 0.650022421648

Experience gain = 0.316689088315

1.(Less than 3) 2, 0, 0

Low

2.(3 to 10) 2, 0, 0

Low

3.(More than 10) 1, 1, 1.0

Career gain = 1.0

1.(Management) 0, 1, 0

High

2.(Service) 1, 0, 0

Low

2.(College) 2, 3, 0.970950594455

Location gain = 0.970950594455

1.(Oregon) 2, 0, 0

Low

2.(California) 0, 3, 0

High

**After pruning:**

Top

Education

1.(High School)

Experience

1.(Less than 3)

Low

2.(3 to 10)

Low

3.(More than 10)

High

2.(College)

Location

1.(Oregon)

Low

2.(California)

High

**Part 2 - KNN**

**Compare (a) & (b): compute the accuracy of predicting test data**

|  |  |  |
| --- | --- | --- |
| k | Without normalizing | With z-score normalization |
| 1 | 0.7518470230334637 | 0.8561495002172969 |
| 5 | 0.7548891786179922 | 0.8700564971751412 |
| 11 | 0.7648848326814428 | 0.878748370273794 |
| 21 | 0.7466318991742721 | 0.8843980877879183 |
| 41 | 0.7522816166883963 | 0.8852672750977836 |
| 61 | 0.7375054324206867 | 0.8826597131681877 |
| 81 | 0.7266405910473707 | 0.877444589308996 |
| 101 | 0.7288135593220338 | 0.8752716210343329 |
| 201 | 0.7314211212516297 | 0.8600608431116906 |
| 401 | 0.7196870925684485 | 0.8396349413298566 |

**(c) Print results in (b) case**

t1 spam, spam, spam, spam, spam, spam, spam, spam, no, no,

t2 spam, spam, spam, spam, spam, spam, spam, spam, no, no,

t3 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t4 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t5 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t6 spam, spam, no, spam, no, no, no, no, spam, spam,

t7 spam, no, no, no, no, no, no, no, no, no,

t8 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t9 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t10 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t11 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t12 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t13 spam, spam, spam, spam, spam, spam, spam, no, no, no,

t14 spam, spam, spam, spam, spam, spam, spam, spam, no, no,

t15 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t16 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t17 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t18 spam, spam, spam, spam, spam, spam, spam, spam, spam, no,

t19 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t20 no, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t21 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t22 spam, spam, spam, spam, spam, spam, spam, no, no, no,

t23 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t24 no, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t25 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t26 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t27 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t28 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t29 spam, spam, spam, spam, spam, spam, spam, spam, no, no,

t30 spam, spam, spam, spam, no, no, no, no, no, no,

t31 spam, no, no, no, no, no, no, no, no, no,

t32 spam, spam, spam, spam, spam, spam, spam, spam, no, no,

t33 spam, spam, spam, spam, spam, no, no, no, no, no,

t34 spam, spam, spam, spam, spam, no, no, no, no, no,

t35 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t36 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t37 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t38 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t39 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t40 no, no, no, no, no, no, no, no, no, no,

t41 no, no, no, no, no, no, no, no, no, no,

t42 spam, spam, spam, spam, spam, spam, spam, spam, no, no,

t43 no, no, no, no, no, no, no, no, no, no,

t44 no, no, no, no, no, no, no, no, no, no,

t45 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t46 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t47 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t48 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t49 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

t50 spam, spam, spam, spam, spam, spam, spam, spam, spam, spam,

(d) **Question: What can you conclude by comparing and examining the KNN performance in (a) and (b)? Give a method to select the optimal k for the KNN algorithm.**

**Answer:** If we use z-score normalization, the accuracy of test data will be much better than the one without normalizing. We can get the best accuracy about 76.5% for k = 11 without normalizing and about 88.5% for k = 41 with z-score normalization. Obviously, the z-score normalization can improve the accuracy of classification. To select the optimal k, we can use the Cross Validation method.

**Part 3 - Use Weka**

PloyKernel exponent = 1

Correctly Classified Instances 717 84.7518 %

Incorrectly Classified Instances 129 15.2482 %

PloyKernel exponent = 2

Correctly Classified Instances 810 95.7447 %

Incorrectly Classified Instances 36 4.2553 %

PloyKernel exponent = 5

Correctly Classified Instances 785 92.7896 %

Incorrectly Classified Instances 61 7.2104 %

PloyKernel exponent = 10

Correctly Classified Instances 678 80.1418 %

Incorrectly Classified Instances 168 19.8582 %

RBFKernel gamma = 0.1

Correctly Classified Instances 670 79.1962 %

Incorrectly Classified Instances 176 20.8038 %

RBFKernel gamma = 1

Correctly Classified Instances 764 90.3073 %

Incorrectly Classified Instances 82 9.6927 %

RBFKernel gamma = 10

Correctly Classified Instances 453 53.5461 %

Incorrectly Classified Instances 393 46.4539 %

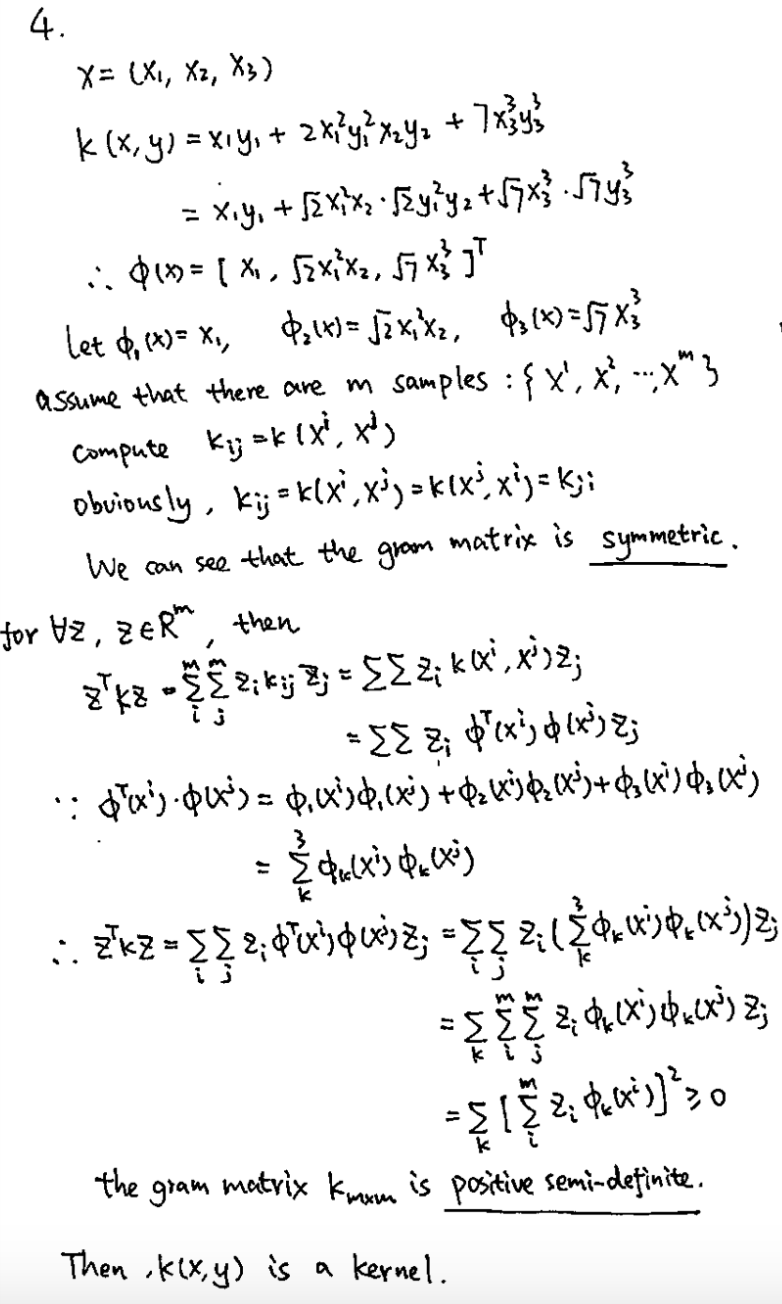
* **Question: Explain why some of the choices do not work well.**

**Answer:** We can find that in PloyKernel, the lowest degree polynomial is the linear kernel, which is not sufficient when a non-linear relationship between features exists. For the data, a degree-2 polynomial is already flexible enough to discriminate between the two classes with a sizable margin. As the exponent parameter grows, the accuracy will decline and the model will be overfitting.

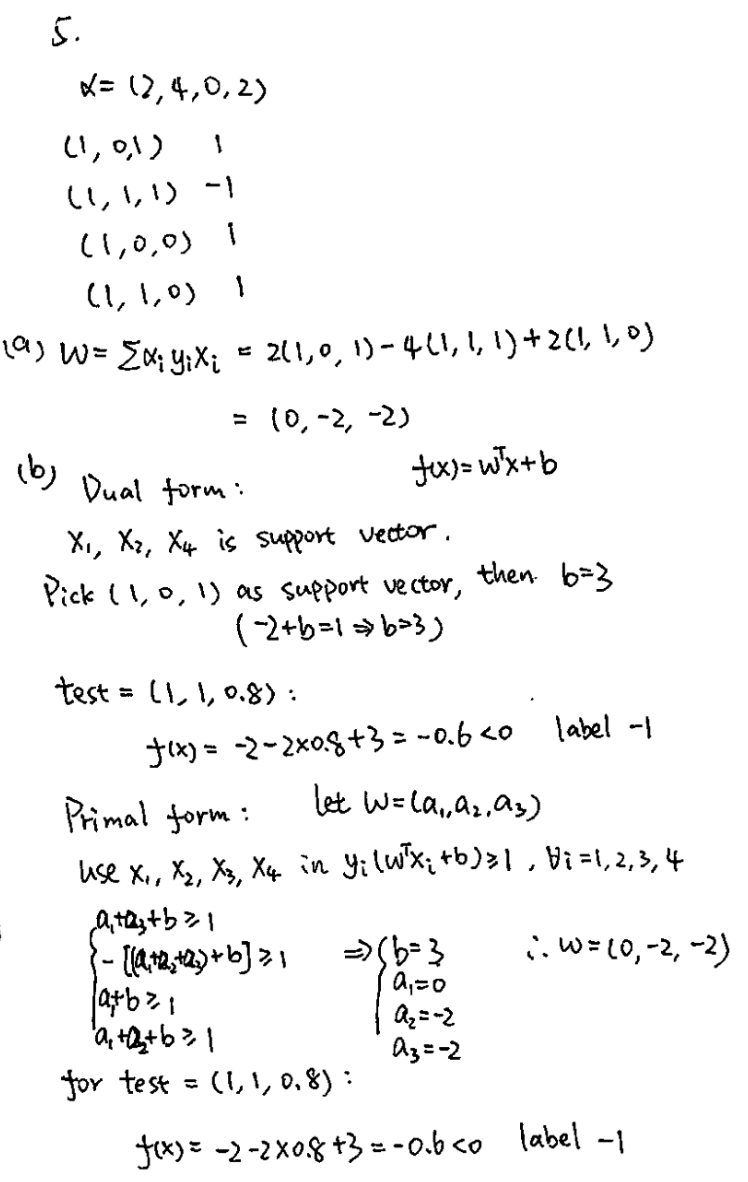
In RBFKernel, the gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. When gamma is very small, the model is too constrained and cannot capture the complexity or “shape” of the data. The region of influence of any selected support vector would include the whole training set. If gamma is too large, the radius of the area of influence of the support vectors only includes the support vector itself and no amount of regularization will be able to prevent overfitting.

As seen from 7 sets of result, the gamma parameter of the RBFKernel and the degree of PolyKernel determine the flexibility of the resulting SVM in fitting the data. If this complexity parameter is too large, overfitting will occur.

**Part 4 Prove Kernel**

****

**Part 5 Dual Form**

****