Ashley Jain
2014CS50280
3 March 2017

COL774: Assignment 2

1.(a)

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \wedge y^{(i)} = 1\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} n_i + |V|}$$

$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \wedge y^{(i)} = 0\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} n_i + |V|}.$$

$$\phi_y = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{m}.$$

Accuracy over training set = 95.3510 %

Accuracy over test set = 93.3303 %

1.(b) **Accuracy** for random prediction = 12.4258 %

Probability of randomly corrected predicted class would be 1/8. (where 8 is no. of categories). So accuracy would be ~100/8%.The algorithm gives almost 8x improvement over random prediction.

Accuracy for majority prediction = 49.4746 %

Most occurred class is acq and no. of acq class in data sets is almost half of the total no. of docs. So accuracy would be ~50%. The algorithm gives almost 2x improvement over random prediction.

1.(c)

Confusion Matrix :-

Predicted Class:- / Actual Class:-	acq	crude	earn	grain	interest	money-fx	ship	trade
acq	1057	26	0	0	0	0	0	0
crude	2	693	1	0	0	0	0	0
earn	4	0	70	0	0	0	0	1
grain	1	6	9	8	0	12	0	0
interest	2	2	3	1	0	2	0	0
money-fx	0	9	4	0	0	108	0	0
ship	2	5	12	0	0	1	37	24
trade	2	2	13	0	0	0	0	70

Highest value of diagonal entry is **acq**. So all **acq** category docs must have many words which are not in other category.

Ship and **trade** are two most confused categories i.e. **ship** category docs are mostly incorrect predicted as **trade** category. So this two category docs would have many common words leads to similar category.

1.(d) Accuracy over training set = 96.6454 %

Accuracy over test set = 94.2896 %

This case accuracies are higher than that of previous case. Since these datas are formed after removing most common words (like the, is, are…) & stopword and stemming. So new datas would be predicted more accurately due to presence of more unique words in each category.

Confusion Matrix :-

Predicted Class:- / Actual Class:-	acq	crude	earn	grain	interest	money-fx	ship	trade
acq	1060	23	0	0	0	0	0	0
crude	5	690	1	0	0	0	0	0
earn	3	0	71	0	0	0	0	1
grain	1	3	9	13	0	10	0	0
interest	3	1	2	0	2	2	0	0
money-fx	0	6	3	0	0	112	0	0
ship	1	3	7	0	0	0	41	29
trade	2	1	9	0	0	0	0	75

2.(a) **Dual SVM Problem**:-

maximize $\Sigma\alpha_i - 0.5*\Sigma\alpha_i\alpha_j y_i y_j (x^{(i)})^T x^{(j)}$ —> $\Sigma\ \alpha_i - 0.5*(\alpha^T*Q*\alpha)$ where Q is $(y*y^T).*(X*X^T)$

such that $0 <= \alpha_i <= C$ and $\Sigma\ y_i\ \alpha_i = 0$

Condition of choosing alpha for support vector:- $\alpha > 10^{-4}$

no. of support vectors = 269

support vector indices = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 ……

2.(b) $b = -0.5*(\max(w^T x)|y=-1 + \min(w^T x)|y=1)$

$W = X^T * (\alpha\ .*\ y) = \Sigma\ \alpha_i y_i x_i$

b = -1.7768

W = [-5.30713404496215 -3.83663055190452 -4.46259656291298 -4.65732083529585 -3.25055520199092 -1.92237929503623 -0.77831836967328……

Accuracy over test set = 61.6667 %

2.(c) Gaussian Kernel matrix :-

$K(i, j) = \exp(-\gamma * (\text{norm}(x_i - x_j))^2)$

no. of support vectors = 258

b = -6.1164

Accuracy over test set = 67.5000 %

This accuracy is higher than that of linear kernel. Since the datas are non-linearly separable, Gaussian kernel transform space into infinite dimension and able to distinguish between non-linearly separable datas more accurately, while linear kernel uses to classify linearly separable datas.

2.(d) Linear Kernel :-

no. of support vectors = 269

Accuracy = 61.6667 %

Gaussian Kernel :-

no. of support vectors = 256

Accuracy = 67.5000 %

Both accuracies are exactly same as that of optimisation method using CVX package. While LibSVM optimises faster than CVX package. So LibSVM selects customised special optimisation problem automatically while CVX package is general purpose for SVM optimisation.

2.(e)

gamma:-	**1**	**10**	**100**	**1000**	**10000**	**100000**	**1000000**
avg. acc. over train (%)	45	45	54.6429	63.9286	66.4286	63.5714	63.5714
accuracy over test (%)	56.6667	56.6667	61.6667	72.5000	75.8333	76.6667	76.6667

Increasing the gamma value, accuracy would be initially increase and further constant. Gamma with value 10^4 gives the best validation accuracy, and with value 10^5 gives best test set accuracy. For large values of C (overfitting), the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified

correctly. While small value of C (underfitting) cause the SVM optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

Graph :-