1 CSE512 Fall 2018 - Machine Learning - Homework 4

Name: Rohit Bhal
Solar ID: 112073893
NetID email address: rbhal@cs.stonybrook.edu / rohit.bhal@.stonybrook.edu

1 Question 1

1.1 Linear Case

Ans For the LOOCV error, the total error is :

$LOOCV = \frac{1}{n} \sum_{i=1}^{n} e_i$, e_i denote the error when i^{th} data point is removed.

Since, the dataset is linearly separable. There are two cases for linear SVM :

1) Removing non-support vector
2) Removing support vector

1) If non-support vector is removed, then the decision boundary doesn't change & the data points won't be misclassified. Thus, $e_i = 0$ for this scenario.

2) If support vector is removed, then the SVM margin will change. In worst case scenario, the example will be misclassified with an error of 1. Thus, removing the support vector will change decision boundary & thus, may add some error to the model. In worst case, all the m support vectors will generate m errors.

Thus, $LOOCV = \frac{1}{n} \sum_{i=1}^{n} e_i \leq \frac{m}{n}$

1.2

If we use a general kernel that separates the data in the high dimensional feature space. Then, the bound on the training error will still hold.

Similarly to the previous answer, if the data is linearly separable in higher dimension. Then, removing non-support vector won't change the SVM margin & thus won't add any error.

On the other hand, if we general kernel to increase dimensions & remove a support vector. Then, the margin in the higher dimension will change. In worst case scenario, it will add 1 error to the model. Thus, for m support vectors ~~in~~ in higher dimensions, the ~~worst~~ worst case scenario would be that each of the vectors is adding one error & are being misclassified by the model.

Thus, $LOOCV = \frac{1}{n} \sum_{i=1}^{n} e_i \leq \frac{m}{n}$.

2 Question 2

1. $H = Y_1 * \text{kernel}(x_1, x_2) * Y_1$, $n \times n$ dimension

$Y_1 = \text{diag}(y) = \text{diag}(Y)$

f = 1-D matrix of length n having -1 as element

$A = [\,]$

$b = [\,]$

$Aeq = Y'$ $1 \times n$ matrix

$beq = 0$ as $\sum_{i=1}^{n} \alpha_i y_i = 0$

lb = 1-D matrix of zero of length n.

ub = 1-D matrix of value $(C*1)$ of length n.

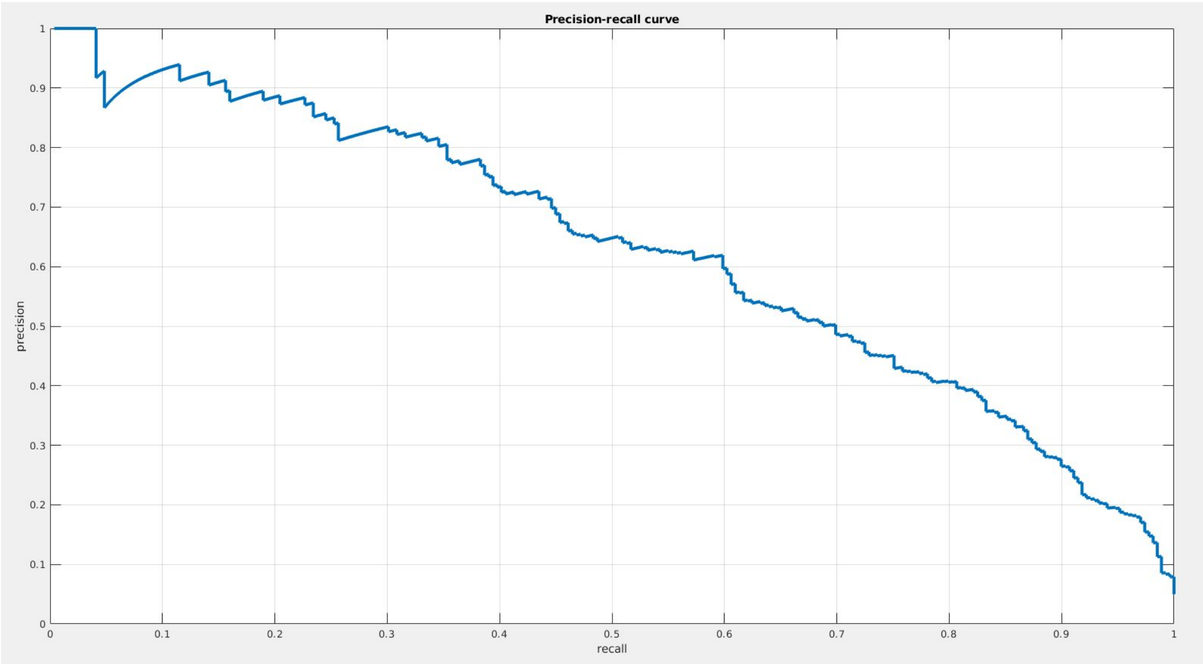
Q 2.4

```
>> [Accuracy, ObjVal, ConfusionMatrix, SupportVector] = trainsvm(valD, valLb, W, B, 0.1)

Accuracy =

    0.9482


ObjVal =

  -10.4788


ConfusionMatrix =

   168    16
     3   180


SupportVector =

   339

>>
```

Q 2.5

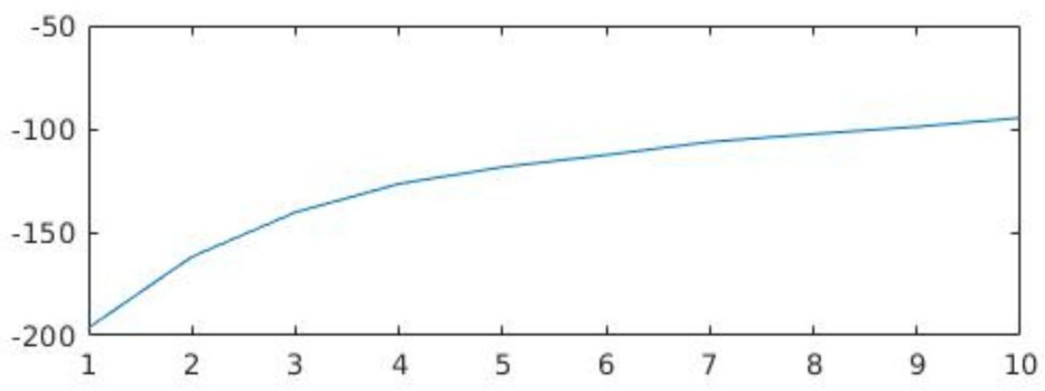
```
>> [Accuracy, ObjVal, ConfusionMatrix, SupportVector] = trainsvm(valD, valLb, W, B, 10)

Accuracy =

    0.9728


ObjVal =

  -4.4360e+03


ConfusionMatrix =

   178     6
     4   179


SupportVector =

   126
```

Q 3.4.1:



Q 3.4.4:

Objective Function graph:



AP Graph: