Cover page for answers.pdf CSE512 Fall 2019 - Machine Learning - Homework 4

Name	Vamshi Muthineni
Solar ID	112607946
NetID email address	vmuthineni@cs.stonybrook.edu
Names of people whom you discussed the homework with	AravindReddy Ravula, Rajireddy Annadi, Mallikarjuna, Meghana, Fathima Cherat.

1.1 & 1.2

(1.1) We know that for a linearly separable set of points

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$b = y - w^T x$$

and we know that $\alpha_i > 0$, if x_i is a support vector

and $\alpha_i = 0$, if not.

hence it is clear that w and b are only dependent on the support vectors and not other points. Hence the hyperplane boundary is determined only by the support vectors.

It is given that there are m support vectors.

LOOCV worst case Analysis :-

Case 1: Removing support vectors.

If removing one support vector causes the hyperplane to shift, then removing 'm' support vectors will give us

$$\text{Error} = \sum_{i=1}^{m} E(sv_i) = m$$

here $E(sv_i) = 1$

Case 2: Removing non-support vectors.

From the initial explanation it is clear that ~~removal~~ of hyperplane doesnot depend on non support vectors. Hence removing such points will never affect the hyperplane in any way.

hence, these points are never misclassified when LOOCV is done.

$\therefore$ Error $= 0$.

$\therefore$ The LOOCV error from the above two cases that

$$\text{LOOCV} = \frac{\text{Error}}{n} = \frac{m}{n}.$$

$\therefore$ LOOCV is bounded by the worst case computation which is $\frac{m}{n}$.

(1.2) It is given that the kernel transforms the data into linearly separable in the high dimension. After transforming the data into linearly separable, it is clear for us that it is an extension of the above proof (problem 1.1)

$\therefore$ Hence the LOOCV bound in previous section holds for the non-linearly separable data with general kernal which makes the data linearly separable in the high dimensional feature space.

2.1

(2.1) The dual objective of kernel svm can be written

as
$$\underset{\alpha}{\text{maximize}} \sum_{j=1}^{n} \alpha_j - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i \alpha_i y_j \alpha_j k(x_i, x_j) \quad \text{—— (1)}$$

$$\text{s.t.} \sum_{j=1}^{n} y_j \alpha_j = 0.$$

$$0 \le \alpha_j \le C \quad \forall j$$

We know that the Quadratic function is

$$\text{minimize } \frac{1}{2} x^T H x + f^T x \quad \text{—— (2)}$$

$$\text{s.t.} \quad Ax \le b$$

$$A_{eq} x = beq.$$

$$lb \le x \le ub.$$

we can make the maximization problem in (1) to match minimization problem in (2) by multiplying with $(-)$.

$$\therefore \underset{\alpha}{\text{minimize}} - \sum_{j=1}^{n} \alpha_j + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i \alpha_i y_j \alpha_j k(x_i, x_j) \quad \text{—— (3)}$$

$$\text{s.t.} \sum_{j=1}^{n} y_j \alpha_j = 0$$

$$0 \le \alpha_j \le C \quad \forall j$$

now compare (2) & (3).

$$x = [\alpha_1 \; \alpha_2 \; \cdots\cdots \; \alpha_n]^T$$

$$A = [\quad] , \; B = [\quad]$$

$$Aeq = [y_1 \; y_2 \; \cdots\cdots \; y_n] , \; beq = 0$$

$$lb = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{n \times 1} \qquad ub = \begin{bmatrix} C \\ C \\ \vdots \\ C \end{bmatrix}_{n \times 1}$$

$$f = \begin{bmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix}$$

$$H = \begin{bmatrix} y_1 y_1 k(x_1, x_1) & y_2 y_1 k(x_2, x_1) & \cdots & y_n y_1 k(x_n, x_1) \\ y_1 y_2 k(x_1, x_2) & y_2 y_2 k(x_2, x_2) & \cdots & y_n y_2 k(x_n, x_2) \\ \vdots & \vdots & & \vdots \\ y_1 y_n k(x_1, x_n) & y_2 y_n k(x_2, x_n) & \cdots & y_n y_n k(x_n, x_n) \end{bmatrix}$$

2.4 At C = 0.1

Objective value is **24.764818**

Validation accuracy is **0.948229**

no of support vectors are **337**

Confusion Matrix:

	Actual Values	
Predicted	180	3
Values	16	168

2.5 At C = 10

Objective value is **112.146132**

Validation accuracy is **0.972752**

no of support vectors are **121**

Confusion Matrix:

	Actual Values	
Predicted	179	4
Values	6	178

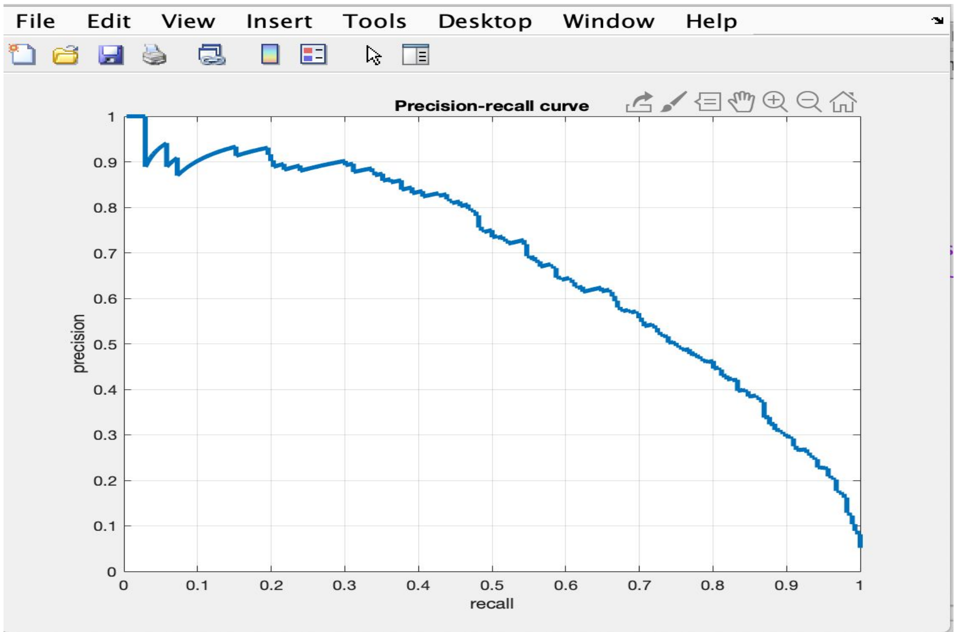
2.6

Test Data Accuracy on Kaggle: **0.50166**

3.4.1

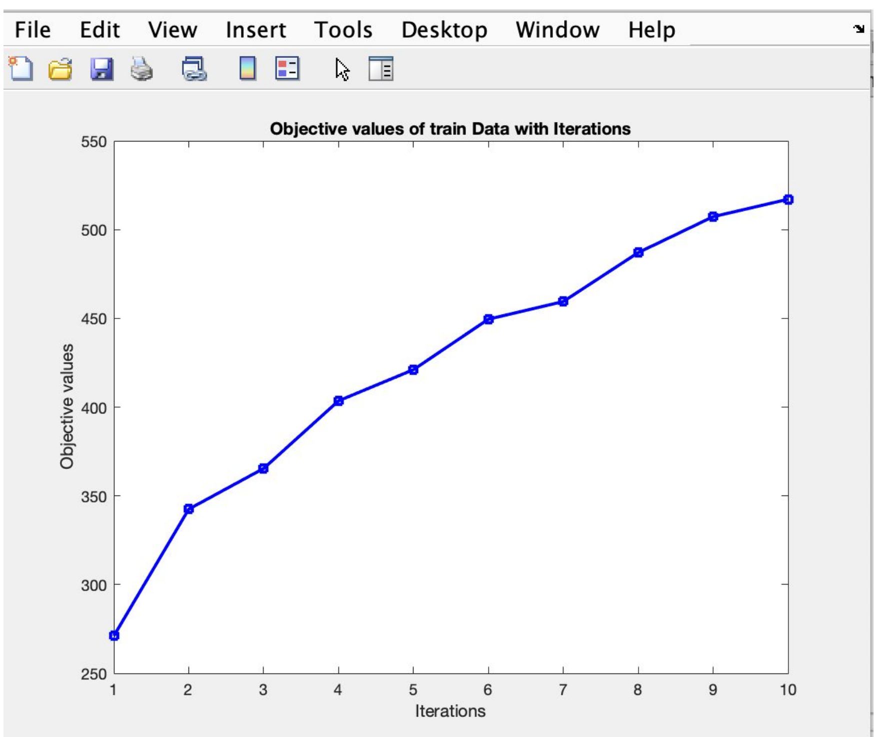
AP(on validation data) at C=2 is **0.6867**

Precision recall curve (on validation data) at C=2.

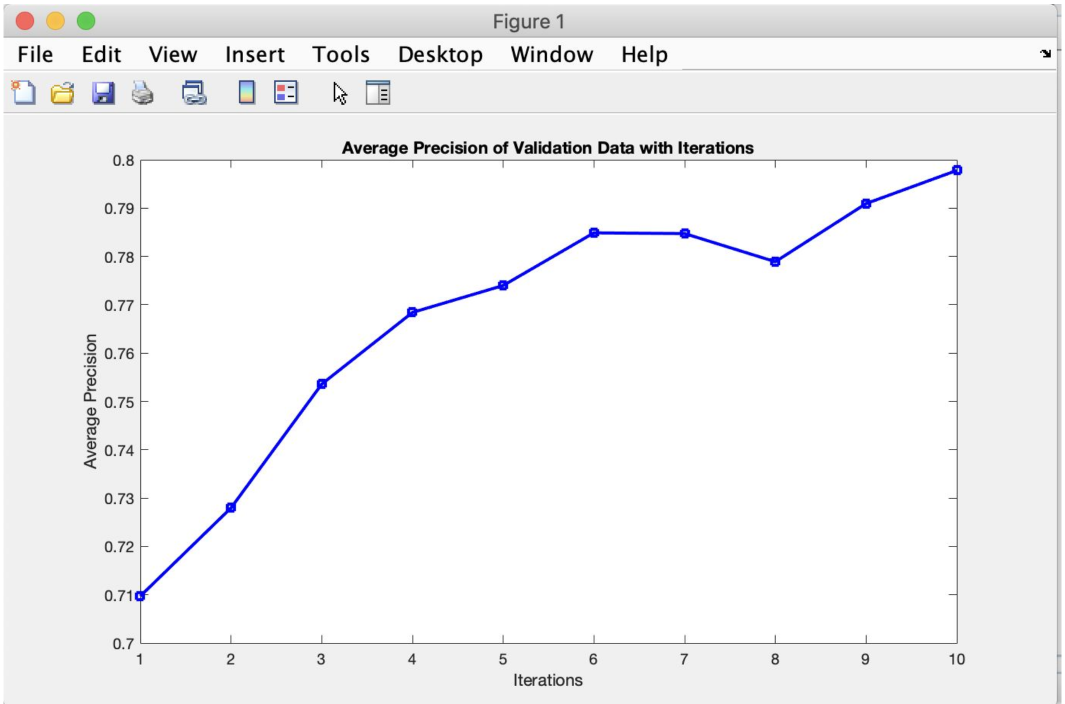


3.4.3

Plot object values (on train data) through 10 iterations at C = 10.



Plot of APs (on validation data) through 10 iterations at C = 10.



3.4.4

AP on test data at C = 10 is **0.7432**.