CPT_S 534 HW2

Yang Zhang

11529139

1. **(a) Show that the decision boundary of the CLOSE classifier is a linear hyperplane of the form sign(w · x + b). Compute the values of w and b in terms of C_+ and C_-.**

   The function of decision boundary should be (any points that lay on decision boundary should have the same distance to C_+ and C_-),

   $$\mathrm{sqrt}[(x-C_+)^2] - \mathrm{sqrt}[(x-C_-)^2]=0,$$

   And then we simplify the above equation into (if distances are the same, so does the distances^2):

   $$(x-C_+)^2 - (x-C_-)^2=0,$$

   Expand it, we got:

   $$(2C_- - 2C_+)\cdot x + (C_+^2 + C_-^2)=0,$$

   Which is the form $w\cdot x + b = 0$, where $w = 2C_- - 2C_+$ and $b = C_+^2 + C_-^2$

   (b) Compute the dual weights (α's). How many of the training examples are support vectors?

$$\begin{cases} C_+ = \frac{1}{n_+}\sum_{i:y_i=1} x_i, \quad C_- = \frac{1}{n_-}\sum_{i:y_i=-1} x_i & ① \\ W = 2C_- - 2C_+ & ② \end{cases}$$

substitude ① into ②:

$$W = \frac{2}{n_+}\sum_{i:y_i=1} x_i + \frac{2}{n_-}\sum_{i:y_i=-1} x_i \quad ③$$

and $W = \sum_{i=1}^{n_+ + n_-} a_i y_i x_i = \sum_{i:y_i=-1} -a_i x_i + \sum_{i:y_i=1} a_i x_i$ ④

③ = ④ we got

$$\frac{2}{n_+}\sum_{i:y_i=1} x_i = \sum_{i:y_i=1} a_i x_i \rightarrow a_i = \frac{2}{n_+} \text{ for } i: y_i=1$$

$$\frac{2}{n_-}\sum_{i:y_i=-1} x_i = \sum_{i:y_i=-1} -a_i x_i \rightarrow a_i = -\frac{2}{n_-} \text{ for } i: y_i=-1$$

Since $a_i \neq 0$ for any i

Therefore, there are $(n_+ + n_-)$ support vectors

2. **Suppose we use the following radial basis function (RBF) kernel: K(x_i , x_j) = exp($-\frac{1}{2}\left\|x_i - x_j\right\|^2$), which has some implicit unknown mapping ϕ(x).**
   (a) Prove that the mapping ϕ(x) corresponding to RBF kernel has infinite dimensions.

   K(x_i , x_j) = exp($-\frac{1}{2}\left\|x_i - x_j\right\|^2$) = exp($-\frac{1}{2}(x_i^2-2x_ix_j + x_j$^2))= exp(-0.5$x_i$^2)*

   exp(-0.5x_j^2)*exp(x_i x_j) = exp(-0.5x_i^2)*exp(-0.5x_j^2)* $\sum_{k=0}^{n} \frac{x_i^k x_j^k}{k!}$

   Therefore the mapping ϕ(x) has infinite dimensions.

   (b) Prove that for any two input examples xi and xj , the squared Euclidean distance of their corresponding points in the higher-dimensional space defined by ϕ is less than 2, i.e., norm[ϕ(xi) − ϕ(xj)]^2 ≤ 2.
   norm[ϕ(xi) − ϕ(xj)]^2 = ϕ(xi)^2 − 2 ϕ(xi)ϕ(xj) + ϕ(xj)^2=K(xi,xi) – 2K(xi,xj) + K(xj,xj)
   norm[ϕ(xi) − ϕ(xj)]^2 = 1-2K(xi,xj)+1
   The value range of K(xi,xj) is [0,1] cause -0.5x^2 $\leq$ 0
   Therefore, norm[ϕ(xi) − ϕ(xj)]^2 ≤ 2

3. **Prove that f(x_{far}; α, b) ≈ b with radial basis function (RBF) kernel: K(x_i , x_j) = exp($-\frac{1}{2}\left\|x_i - x_j\right\|^2$), which has some implicit unknown mapping ϕ(x).**

   The distance between x_{far} and any training data x_i is very large which implies that exp($-\frac{1}{2}\left\|x_{far} - x_i\right\|^2$) is very small (closed to zero), so that:

   f(x_{far}; α, b) = $\sum_{i=0}^{n} y_i a_i \exp(-\frac{1}{2}\left\|x_{far} - x_i\right\|^2) + b$ = very small value + b ≈ b

4. **The function K(x_i , x_j) = − <x_i , x_j> is a valid kernel. Prove or Disprove it**

   The function K(x_i , x_j) = − <x_i , x_j> is not a valid kernel.

   We know that K'(x_i , x_j) = <x_i , x_j> is a valid kernel, so that based on positivity, K' follows:

   $$t^T K' t \geq 0$$

   However, K=-K' implies that:

   $$t^T K t = -t^T K' t \leq 0$$

   which violates the positivity property, therefore, K(x_i , x_j) = − <x_i , x_j> is not a valid kernel.

5. **You are provided with n training examples: (x1, y1),(x2, y2), · · · ,(xn, yn,), where xi is the input example, yi is the class label (+1 or -1). The teacher gave you some additional information by specifying the costs for different mistakes C+ and C−, where C+ and C− stand for the cost of misclassifying a positive and negative example respectively. a. How will you modify the Soft-**

margin SVM formulation to be able to leverage this extra information? Please justify your answer.

To leverage the extra information, we modify the original objective function:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \ s.t.\ y_i(w \cdot x_i + b) \geq 1$$

to soft margin objective function:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + c \sum_{i=1}^{n} \xi_i \ s.t.\ y_i(w \cdot x_i + b) \geq 1, \xi_i \geq 0$$

$$\textbf{where c} = \begin{cases} C_- & negative\ x_i\ is\ incorrectly\ classified \\ C_+ & positive\ x_i\ is\ incorrectly\ classified \\ 0 & x_i\ is\ correctly\ classified \end{cases}$$

In this way, if the training sample was correctly classified, c would be zero, so no penalty would be applied. If a positive training sample was incorrectly classified as negative class, then the Lagrange multiplier would be C_+ corresponding the cost of misclassifying a positive training sample. Vice versa, if a negative training sample was incorrectly classified as positive class, the cost C_+ would be applied.

6. **Consider the following setting. You are provided with n training examples: (x1, y1, h1),(x2, y2, h2), · · · ,(xn, yn, hn), where xi is the input example, yi is the class label (+1 or -1), and hi > 0 is the importance weight of the example. The teacher gave you some additional information by specifying the importance of each training example.**
   (a) **How will you modify the Soft-margin SVM formulation to be able to leverage this extra information? Please justify your answer.**

   To leverage the extra information, we modify the original objective function:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \ s.t.\ y_i(w \cdot x_i + b) \geq 1$$

   to soft margin objective function:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + h_i \sum_{i=1}^{n} \xi_i \ s.t.\ y_i(w \cdot x_i + b) \geq 1, \xi_i \geq 0$$

   In this way, if the training sample was correctly classified, c would be zero, so no penalty would be applied. If a training sample was incorrectly classified, then the Lagrange multiplier would be h_i corresponding the how importance is training sample.

   (b) **How can you solve this learning problem using the standard SVM training algorithm? Please justify your answer**

   By applying standard SVM, we can modify the data input format from (x_i, y_i, h_i) into ($x_i h_i$, y_i). In this way, if the training sample is in support vector set, then its Lagrange multiplier would be a_i.*h_i, which corresponds to its importance.

7.

(a) To construct the training formulation, we can divided all those clusters into cluster pairs. each pair contains 1 positive cluster and 1 negative cluster. We denotes the ith pair to be P_i, for each P_i we want maximize the margin which is same as:

$$\min_{w_i, b_i} \frac{1}{2} \|w_i\|^2 \text{ s.t. } y_i (w_i \cdot x_i + b_i) \geq 1$$

we can apply the concept of CLOSE classifier from Q1 by defining C_+^i to be center of possitive cluster in P_i, verse visa C_-^i to be the center of negative cluster in P_i

Thus:

$$w_i = 2C_-^i - 2C_+^i, \quad b_i = C_+^2 + C_-^2$$

Therefore the constrain for the objective function turns to

$$y_i((2C_-^i - 2C_+^i) \cdot x_i + C_+^2 + C_-^2) \geq 1$$

(b) To refine the clusters, we can introduce a importance parameter M_i each pair P_i. There are many way to assign value to M_i, one easy way is to let M_i be the ratio of samples in pair P_i over total sample, i.e.

$$M_i = (n_+^i + n_-^i) / \sum_k (n_+^k + n_-^k)$$

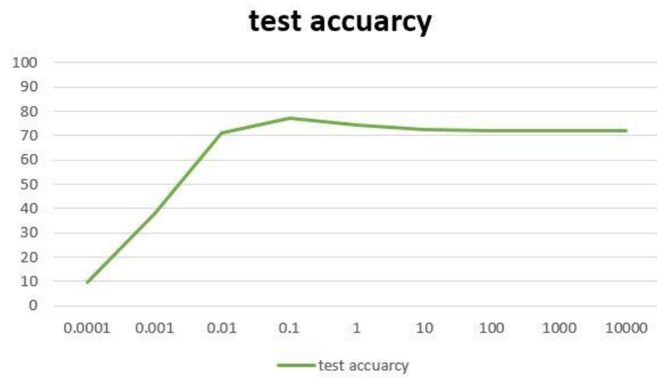
To apply this, we can merge M_i into the discriminant function

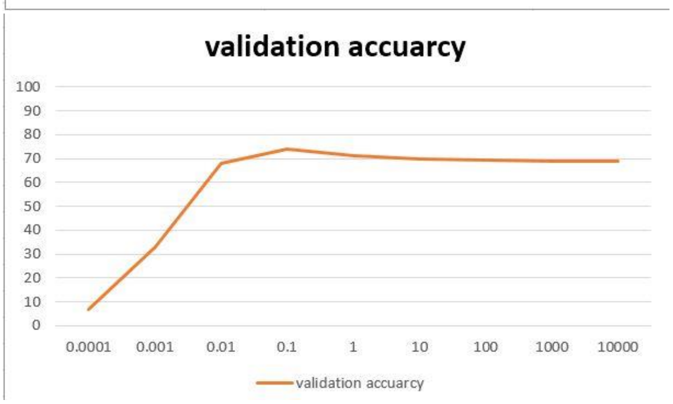
$$f(x) = \sum_i f_i(x) \cdot M_i$$

where $f_i(x)$ is the classifier trained from pair P_i

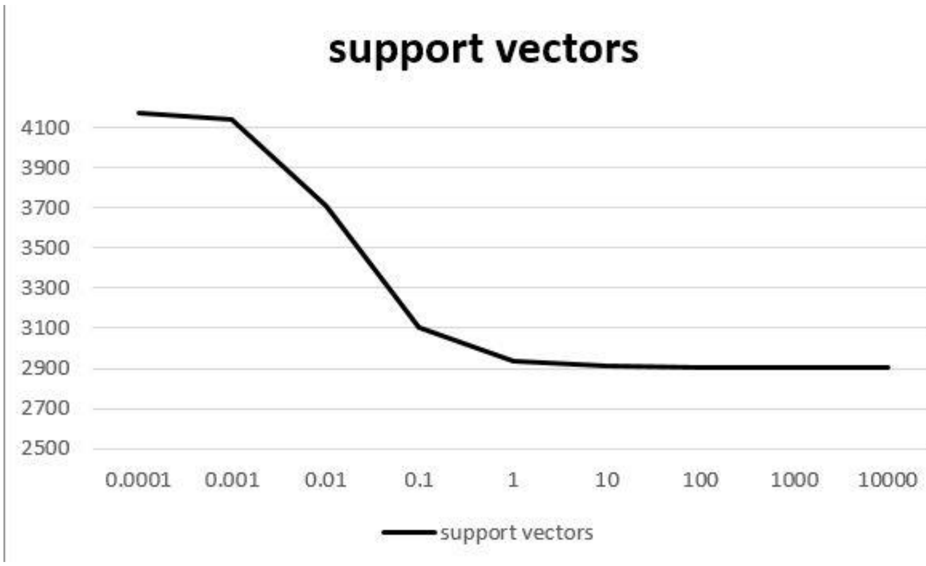
(c) The stopping criteria is either runing over all pair or making the current cumulated $f(x)$ convergence

8. (a)





For the training accuracy, it always increases with higher c value. While for validation and testing accuracy, it increases with higher c value, and reaches max accuracy at c = 0.1, then the accuracy starts falling down.



For support vectors, the number of SV increasing with higher value of c, and finally meet the convergence.

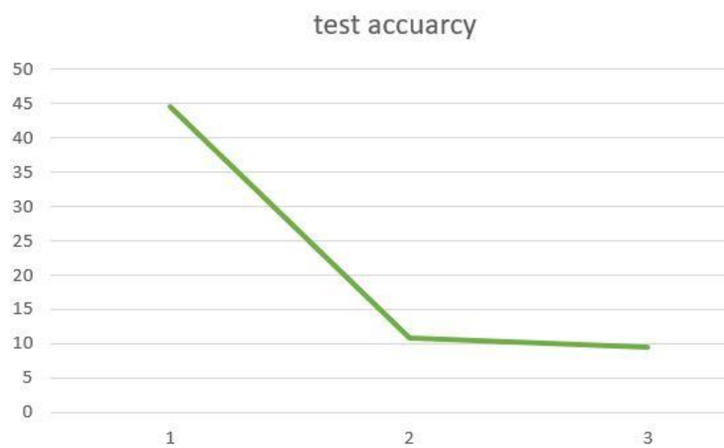
(b)

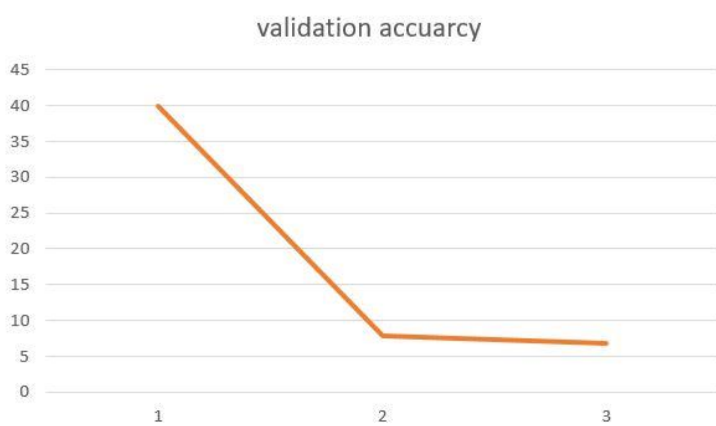
The accuracy of the SVM trained by combined set of training and validation samples is 78.25%

The confusion matrix:

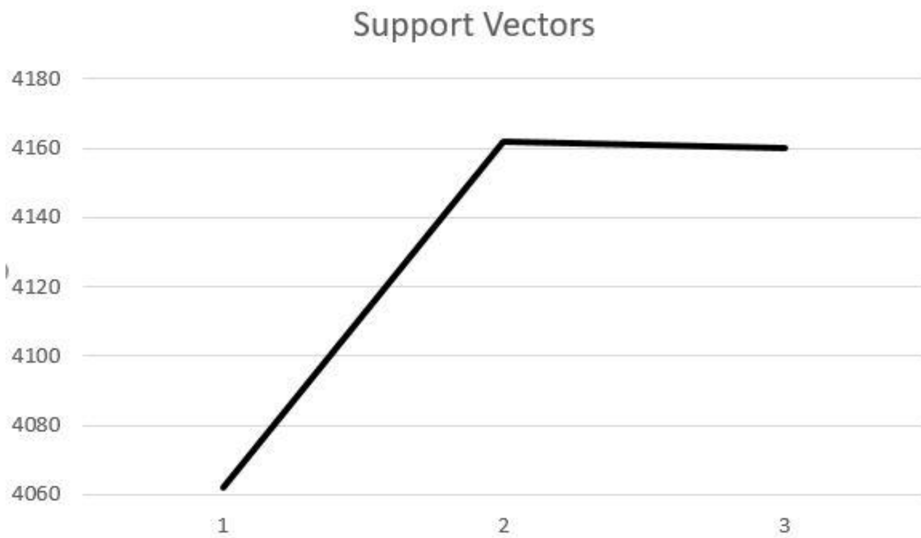
	~a	~b	~c	~d	~e	~f	~g	~h	~i	~j	~k	~l	~m	~n	~o	~p	~q	~r	~s	~t	~u	~v	~w	~x	~y	~z
a	43	14	7	14	40	2	16	7	26	0	7	40	9	33	39	7	1	31	9	10	16	0	2	0	6	7
b	10	2	9	6	15	1	4	1	8	0	6	8	0	11	7	1	1	12	4	4	9	2	0	3	0	0
c	14	3	26	4	13	2	8	0	9	0	1	14	4	19	32	3	0	14	3	8	9	2	1	0	1	2
d	10	3	2	14	9	2	11	0	11	0	1	5	1	6	5	7	0	9	6	2	11	0	0	0	3	2
e	41	18	9	19	58	6	20	7	21	1	7	15	24	38	34	17	0	28	8	11	33	5	1	1	5	8
f	2	1	2	0	7	0	6	2	1	0	0	7	1	6	19	4	1	2	0	3	6	1	0	0	4	2
g	14	3	7	2	40	5	14	3	21	2	3	14	10	19	18	8	1	15	2	8	5	3	0	0	3	3
h	8	0	2	2	6	0	2	0	4	0	0	5	4	11	15	2	0	2	1	0	4	0	0	0	0	4
i	44	7	10	11	36	12	27	5	36	0	18	29	8	69	27	13	3	28	11	19	22	3	3	1	9	6
j	0	0	1	1	0	0	3	1	1	0	0	0	0	2	1	0	0	1	0	0	2	0	0	0	0	0
k	3	3	7	8	15	1	4	1	4	0	3	2	3	3	15	4	0	2	1	4	3	0	1	3	0	1
l	33	7	21	4	22	3	15	2	38	0	2	26	6	19	27	3	0	19	2	3	18	0	0	1	6	5
m	17	3	4	2	21	3	8	0	16	1	0	10	28	11	7	1	0	5	2	2	3	0	0	0	2	1
n	35	9	20	7	33	6	37	7	32	0	7	27	22	40	39	10	2	35	12	15	12	3	3	2	11	6
o	19	9	10	12	28	4	28	1	37	1	1	21	10	46	33	4	2	26	1	10	13	7	4	2	8	4
p	5	0	1	3	10	6	12	0	9	0	1	6	3	18	4	12	0	14	13	2	4	3	1	2	2	2
q	3	0	3	2	2	1	0	1	3	0	0	1	0	5	1	0	1	1	0	0	3	1	1	1	0	0
r	29	4	8	3	15	1	5	0	13	0	2	13	7	17	27	15	1	9	2	9	15	6	9	5	3	7
s	11	2	7	4	13	1	8	0	20	0	2	7	1	18	11	1	0	7	3	3	4	3	1	0	3	0
t	15	6	9	3	15	2	13	1	14	0	2	9	5	8	26	5	0	11	4	2	5	5	1	1	3	2
u	16	3	13	5	18	2	17	4	19	1	6	16	0	31	15	5	0	6	1	10	6	0	0	0	5	1
v	3	1	1	0	3	0	4	1	11	0	3	9	0	4	4	0	0	3	0	2	11	3	0	0	0	3
w	4	2	3	2	12	2	0	0	6	0	1	2	0	1	2	0	0	4	0	1	2	0	0	0	0	1
x	4	1	0	0	1	1	3	3	3	0	0	3	0	6	0	0	1	1	10	0	4	0	0	9	0	1
y	5	1	6	0	11	1	8	0	8	0	0	9	0	19	2	8	1	9	0	5	1	1	1	0	2	3
z	7	0	2	5	6	0	2	0	13	0	2	7	2	8	3	0	0	1	8	4	3	0	1	2	0	1

(c)





For all the three test, all accuracies decrease with higher degree of polynomial, and seems to meet convergence after degree of 3.



For support vectors, the number of support vectors increases with higher degree. And it seems to meet convergence at degree of 2.