CPT_S 570: Homework #2

Instructor: Jana Doppa

Sheng Guan

Problem 1

Answer:
a): CLOSE classifies a test example x by assigning it to the class whose center is closest. The decision boundary is sign(Dis(x, C_-)Dis(x, C_+)- Dis(x, C_+))

$$\begin{aligned} sign(Dis(x, C_-) - Dis(x, C_+)) &= sign((x - C_-)^2 - (x - C_+)^2) \\ &= sign(2C_+x - 2C_-x + C_-^2 - C_+^2) \\ &= sign((2C_+ - 2C_-)x + (C_-^2 - C_+^2)) \\ &= sign(\omega x + b) \end{aligned} \tag{1}$$

Hence, we can get $\omega = 2C_+ - 2C_-$
$b = \|C_-\|^2 - \|C_+\|^2$
b)

$$\begin{aligned} \omega &= \frac{2}{n_+} \sum_{i:y_i=1} x_i - \frac{2}{n_-} \sum_{i:y_i=-1} x_i \\ &= \frac{2}{n_+} \sum_{y_i=1} y_i x_i + \frac{2}{n_-} \sum_{y_i=-1} y_i x_i \end{aligned} \tag{2}$$

$$\alpha_i = \begin{cases} \frac{2}{n^+} & \text{if } y_i = 1 \\ \frac{2}{n^-} & \text{if } y_i = -1 \end{cases}$$

Every training example with a positive or negative class has a non-zero alpha value , so the number of support vectors equals to $n_+ + n_-$

Problem 2

Answer:
a):

$$< \phi(x_i), \phi(x_j) >= K(x_i, x_j) = exp(-\frac{1}{2}\left\|x_i - x_j\right\|^2) = exp(-\frac{1}{2}\left\|x_i\right\|^2) \bullet exp(-\frac{1}{2}\left\|x_j\right\|^2) \bullet exp(x_i^T x_j) \tag{3}$$

If we use taylor series to replace the last item we can get:

$$< \phi(x_i), \phi(x_j) >= K(x_i, x_j) = exp(-\frac{1}{2}\left\|x_i\right\|^2) \bullet exp(-\frac{1}{2}\left\|x_j\right\|^2) \bullet \sum_{k=0}^{\infty} \frac{(x_i^T x_j)^k}{k!} \tag{4}$$

for $\sum_{k=0}^{\infty} \frac{(x_i^T x_j)^k}{k!}$, as $k \in 0...\infty$, it is infinite, thus RBF kernel has infinite dimensions.

b):
From $K(x_i, x_j) = exp(-\frac{1}{2}\left\|x_i - x_j\right\|^2)$
we know that: 1) $K(x_i, x_i) = 1$, 2)$K(x_i, x_j) > 0$

$$\begin{aligned} \left\|\phi(x_i) - \phi(x_j)\right\|^2 &=< \phi(x_i), \phi(x_i) > + < \phi(x_j), \phi(x_j) > -2 < \phi(x_i), \phi(x_j) > \\ &= K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j) = 2 - 2K(x_i, x_j) < 2 \end{aligned} \tag{5}$$

Problem 3

Answer:
As x_{far} is far from $x_i \in SV$, we know that $\left\|x_i - x_{far}\right\| \to \infty$, thus RBF kernel $K(x_i, x_{far}) = exp(-\frac{1}{2}\left\|x_i - x_{far}\right\|^2) \to 0$,
$f(x_{far}; \alpha, b) = \sum_{i \in SV} y_i \alpha_i K(x_i, x_{far}) + b \approx b$

Problem 4

Answer:
It is not a valid kernel. Proof for the above statement is as below:
For symmetry: $K(x_i, x_j) = - < x_i, x_j >= - < x_j, x_i >= K(x_j, x_i)$, it works;
For positive semi-definite: $K(x_i, x_i) = - < x_i, x_i >= -\left\|x_i^2\right\| \leq 0$, here we consider finite set of m points, let t^T be a $1 \times m$ vector, then $t = [1, 0, 0...0]^T$,$t^T K t = K(x_1, x_1) \leq 0$, as it does not satisfy positive semi-definite, it's not a valid kernel.

Problem 5

Answer:
We will weight the positive errors with C_+ and weight the negative errors with C_- Now we have:

$$\begin{aligned} & min_{w,b} \frac{1}{2}\left\|w\right\|^2 + C(\sum_{i:y_i=1} C_+ \xi_i + \sum_{i:y_i=-1} C_- \xi_i) \\ & subject \quad to \quad y^i(w \bullet x^i + b) \geq 1 - \xi_i, \quad i = 1, ..., N \\ & \xi_i \geq 0, i = 1, ..., N \end{aligned} \tag{6}$$

Problem 6

Answer:
a)If we give each example an importance weight, we should penalize more if we have errors for important example as we want to make these examples correctly classified. We will weight the errors based on the importance weights. Now we have:

$$\begin{aligned} & min_{w,b} \frac{1}{2}\left\|w\right\|^2 + C\sum_{i=1}^{N} \xi_i h_i \\ & subject \quad to \quad y^i(w \bullet x^i + b) \geq 1 - \xi_i, \quad i = 1, ..., N \\ & \xi_i \geq 0, i = 1, ..., N \end{aligned} \tag{7}$$

b)By using the standard SVM training algorithm, we can just duplicate copies of each example equal to the weight h_i times.

Problem 7

Answer:
a)Suppose now we have $k+$ positive clusters and $k-$negative clusters, and each cluster contains a set of instances who have the same label. Now We just use the center of each cluster as a representative point, in

this case we will have $k+$ positive points and $k-$ negative points.First, we give some definitions:
The center x_{center} of a cluster:

$$x_{center} = \frac{\sum_{i=1}^{N} x_i}{N} \tag{8}$$

And the plain SVM algorithm can solve this problem. Also as Problem 6 indicates, we can further use the size of each cluster as the importance weight h. And finally the problem becomes:

$$\begin{aligned} & min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i h_i \\ & subject \quad to \quad y^i(w \bullet x^i + b) \geq 1 - \xi_i, \quad i = 1, ..., k_+ + k_- \\ & \xi_i \geq 0, i = 1, ..., k_+ + k_- \end{aligned} \tag{9}$$

b)By solving the previous problem, we can get the classifier $w \bullet x^i + b$
Now we need to make a decision whether to refine the classifier for each cluster to make it more accurate. If all points in a cluster can be classified correctly by the current classifier, then we won't further split the cluster. To do this, we can scan over each point in the cluster and use the classifier $w \bullet x^i + b$ to check whether $y^i(w \bullet x^i + b) \geq 1$ using the hard margin. If the above condition is satisfied for every point in the cluster, then we will not split. Otherwise, we will split the cluster into two sub-clusters: one cluster which contains all the points that are classified correctly. And the other cluster which contains all the points that are not classified correctly. We can apply the plain SVM algorithm on this set of examples again to get a new sub-classifier $w' \bullet x^i + b'$.
c)For each iteration, we use the current classifier(s) for each cluster to check whether for each point, $y^i(w \bullet x^i + b) \geq 1$ satisfies.
the termination criteria can be: no more clusters will be added to the refining step. Optional question): For highly non-separable dataset, this method will continue going to the lower level to split the cluster. In the end our solution for the coarse problem won't apply on the most fine problems and this solution will fail.

Problem 8

Answer:
a)An approach is we determine this set of support vectors from SV according to the validation accuracy performance since the training size n is very large. We can random select a portion n' as validation data. After we get all the SV, we can apply our model on the validation data on C_N^B, N is the total number of SV, and according to the corresponding validation accuracy to select the best combination (if there's a tie, then random choose one). Then we can get the B support vectors from SV.

b) We can add one more step to remove the support vectors from the kernel expansion. – Budget Kernel Perceptron. In step 7, we remove the support vector which leads the maximum of $argmax_{i \in S} y_i(ytt(x_i) - \alpha_i K(x_i, x_i))$ to reduce the SV size by removing the least significant support vector.

Algorithm 1 Budget Kernel Perceptron(B)

procedure MyProcedure
  $S \leftarrow \emptyset, b \leftarrow 0$.
  Pick a random example (x_t, y_t)
  Compute $ytt(x_t) = sum_{i \in S} \alpha_i K(x_t, x_i) + b$
  if $y_t * ytt(x_t) \leq 0$ **then**,
    $S \leftarrow S \cup t, \alpha_t \leftarrow \alpha_t + y_t$

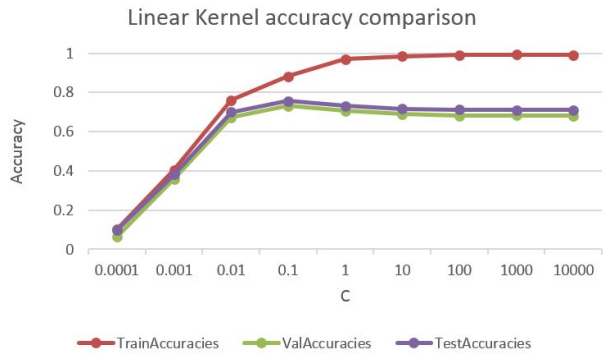
  if $\|S\| > B$ **then**,

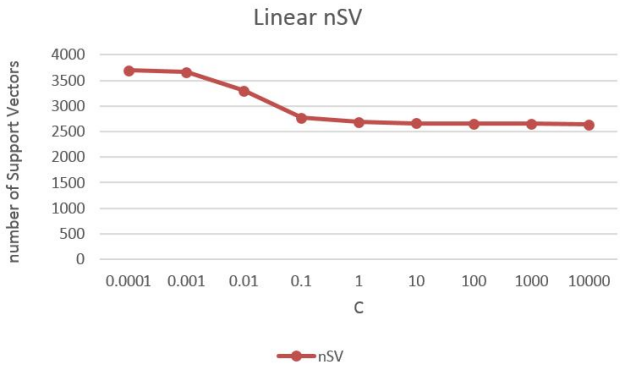
    $S \leftarrow S - \arg\max_{i \in S} y_i(ytt(x_i) - \alpha_i K(x_i, x_i))$
  EndIf
  EndIf
  Return to step 3.

Problem 9

Answer:

a)Three curves and the number of support vectors as C changes are shown as:





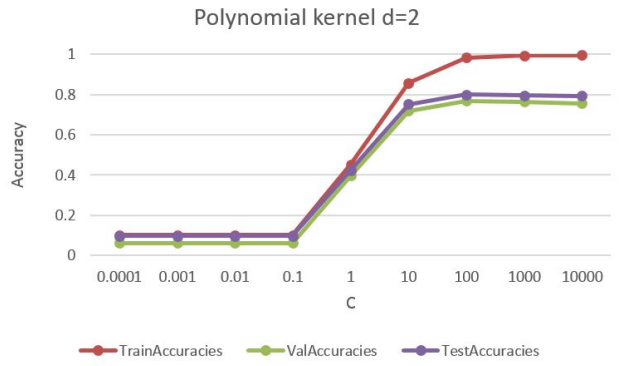
Clearly,the number of support vectors cannot guarantee a high accuracy performance. In this comparison, at the beginning, more examples are considered as support vectors, but the accuracies are very low. The training accuracy performance is much better than validation and testing accuracy performance.
b)Based on a)'s result, we choose C=0.1. The testing accuracy is 77.528% and the confusion matrix is shown as the below picture:

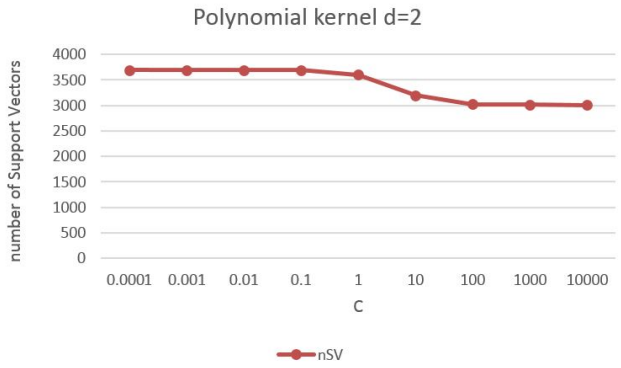
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	2908	4	13	78	153	1	21	5	1	0	15	1	50	195	118	0	3	3	14	3	36	0	2	8	2	14
2	31	917	0	61	6	3	7	25	1	1	8	8	0	25	24	2	1	1	12	2	14	0	0	3	3	5
3	44	1	1625	1	60	1	1	1	2	0	0	11	0	5	44	3	0	109	0	0	10	2	1	0	1	0
4	106	24	0	1018	16	6	22	5	0	3	10	5	0	17	42	0	3	0	2	6	6	1	1	3	9	17
5	178	1	94	5	3789	18	14	2	1	0	11	3	24	41	73	64	0	102	2	36	11	1	1	6	2	41
6	2	4	1	3	34	435	13	0	6	0	6	13	1	5	5	39	1	129	11	125	0	0	0	1	7	3
7	78	5	6	47	43	8	1691	3	6	2	6	7	2	11	46	23	47	21	89	3	16	6	1	3	67	12
8	17	98	2	46	5	3	0	405	4	0	73	15	0	81	1	0	0	0	4	5	26	1	0	2	0	1
9	7	1	25	0	10	1	5	4	3593	2	5	732	0	2	0	1	1	30	5	16	1	0	0	7	8	0
10	0	1	0	15	0	0	13	0	45	78	0	7	0	0	1	0	0	0	4	5	2	2	0	0	2	1
11	22	13	3	15	97	1	0	81	0	0	434	12	4	27	1	0	0	30	2	13	37	1	0	10	1	14
12	0	2	109	9	2	3	6	13	140	0	8	2485	0	0	0	13	0	14	1	14	6	3	0	0	29	1
13	81	1	0	1	16	0	0	1	0	0	1	0	1224	112	0	0	0	5	0	2	1	0	9	1	0	0
14	265	5	2	22	73	1	17	7	0	0	19	0	68	3887	51	8	0	6	3	0	86	2	45	8	10	7
15	83	3	15	13	33	0	22	1	0	0	0	0	7	40	3277	2	2	3	3	1	38	3	0	1	6	3
16	4	1	0	0	13	39	16	0	0	0	1	6	2	13	8	1024	15	24	1	51	1	0	0	1	22	2
17	15	5	0	3	5	3	95	1	1	0	2	6	0	3	5	18	100	3	15	18	1	0	1	1	9	1
18	38	3	19	3	90	57	5	1	9	0	17	31	1	33	0	33	0	1991	3	34	5	31	0	11	26	7
19	33	17	5	6	16	19	124	1	11	5	0	20	4	9	6	0	2	34	919	18	0	0	0	4	8	3
20	29	9	3	39	73	40	50	13	30	0	12	113	4	27	0	1	0	85	19	1327	8	20	0	3	46	18
21	203	2	4	10	31	0	3	6	2	0	7	0	4	93	80	0	1	2	4	5	1804	88	12	1	0	0
22	4	2	1	4	8	2	5	2	0	0	3	1	0	7	7	1	0	21	13	6	166	338	4	0	2	1
23	12	0	0	4	18	0	0	0	0	0	0	0	12	36	3	0	0	0	0	1	46	3	340	0	0	0
24	22	0	0	0	10	4	3	1	0	0	10	0	6	20	0	1	0	12	9	9	2	11	0	202	21	19
25	6	1	0	23	4	5	335	2	4	2	4	53	1	14	9	5	4	36	27	58	4	12	1	1	500	9
26	25	2	3	8	137	2	43	2	10	0	22	11	1	11	2	2	0	20	18	76	0	3	0	44	33	542

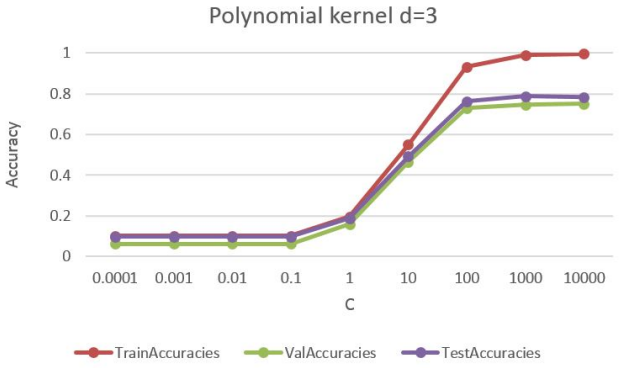
if we consider the number of examples with the ground truth as denominator, the confusion matrix can be further represented as: (worth to mention that lots of 0.00 in the picture is due to the small molecule.)

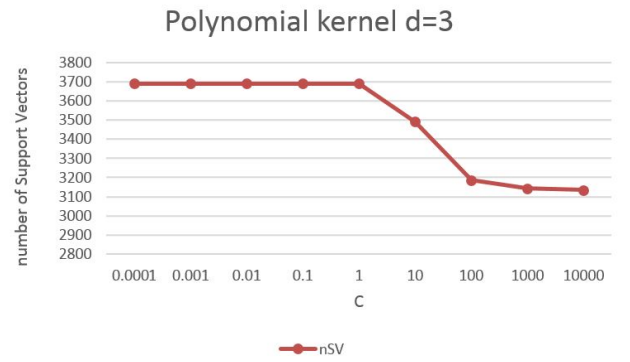
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	0.80	0.03	0.02	0.08	0.04	0.00	0.03	0.02	0.00	0.00	0.03	0.00	0.06	0.06	0.02	0.00	0.05	0.02	0.03	0.01	0.09	0.01	0.03	0.06	0.01	0.02
1	0.00	0.79	0.00	0.02	0.00	0.00	0.00	0.12	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.85	0.00	0.02	0.00	0.00	0.00	0.01	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.02	0.05	0.00	0.77	0.00	0.00	0.02	0.06	0.00	0.09	0.02	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.02	0.00	0.01	0.01	0.00	0.02	0.01
4	0.04	0.01	0.03	0.01	0.84	0.04	0.02	0.01	0.00	0.00	0.12	0.00	0.01	0.02	0.01	0.01	0.02	0.04	0.01	0.04	0.01	0.01	0.04	0.03	0.00	0.13
5	0.00	0.00	0.00	0.00	0.00	0.52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.01	0.02	0.02	0.02	0.00	0.00	0.00	0.01	0.00	0.00
6	0.01	0.01	0.00	0.02	0.00	0.02	0.75	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.01	0.01	0.31	0.00	0.10	0.03	0.00	0.01	0.00	0.01	0.30	0.04
7	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.51	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.81	0.26	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.01
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.01	0.00	0.01	0.00	0.01	0.00	0.09	0.00	0.00	0.53	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.01	0.00	0.03	0.00	0.02
11	0.00	0.01	0.01	0.00	0.00	0.02	0.00	0.02	0.16	0.04	0.01	0.87	0.00	0.00	0.00	0.00	0.02	0.01	0.02	0.06	0.00	0.00	0.00	0.00	0.05	0.01
12	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.84	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.02	0.00	0.00
13	0.05	0.02	0.00	0.01	0.01	0.01	0.00	0.10	0.00	0.00	0.03	0.00	0.08	0.85	0.01	0.01	0.01	0.01	0.01	0.01	0.04	0.01	0.08	0.06	0.01	0.01
14	0.03	0.02	0.02	0.03	0.02	0.01	0.02	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.92	0.01	0.02	0.00	0.00	0.03	0.01	0.01	0.00	0.01	0.00	0.00
15	0.00	0.00	0.00	0.00	0.01	0.05	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.82	0.06	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
17	0.00	0.00	0.06	0.00	0.02	0.15	0.01	0.00	0.01	0.00	0.04	0.00	0.00	0.00	0.00	0.02	0.01	0.81	0.03	0.04	0.00	0.04	0.00	0.03	0.03	0.02
18	0.00	0.01	0.00	0.00	0.00	0.01	0.04	0.01	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.73	0.01	0.00	0.02	0.00	0.02	0.02	0.02
19	0.00	0.00	0.00	0.00	0.01	0.15	0.00	0.01	0.00	0.03	0.02	0.00	0.00	0.00	0.00	0.04	0.06	0.01	0.01	0.67	0.00	0.01	0.00	0.02	0.05	0.07
20	0.01	0.01	0.01	0.00	0.00	0.00	0.01	0.03	0.00	0.01	0.05	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.76	0.28	0.10	0.01	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.04	0.57	0.01	0.03	0.01	0.00
22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.72	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.56	0.00	0.04
24	0.00	0.00	0.00	0.01	0.00	0.01	0.03	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.02	0.03	0.01	0.01	0.02	0.00	0.00	0.00	0.06	0.45	0.03
25	0.00	0.00	0.00	0.01	0.01	0.00	0.01	0.00	0.00	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.05	0.01	0.53

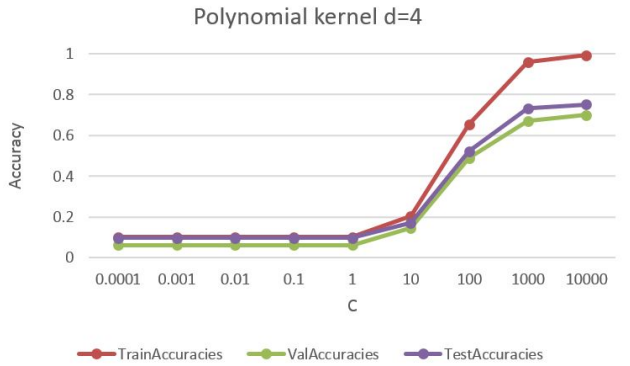
c) The result for polynomial kernel with different degrees is shown as follows:

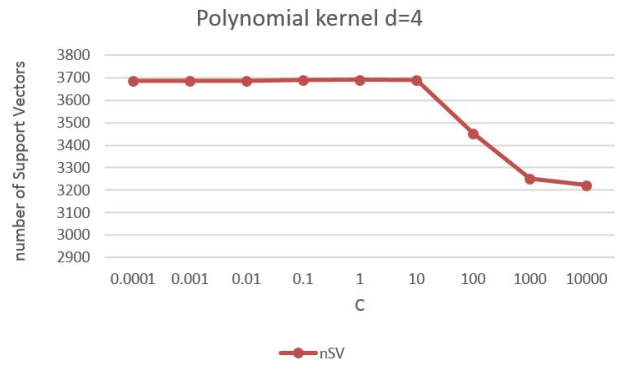


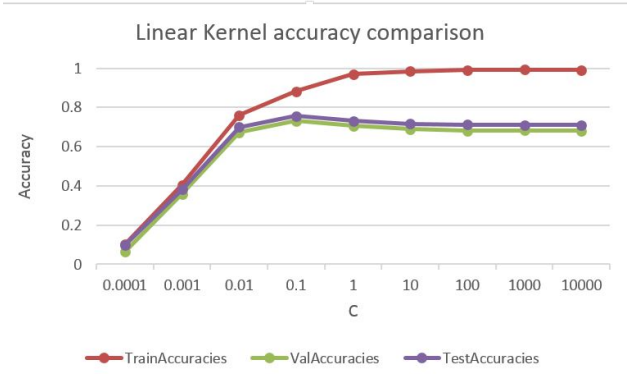


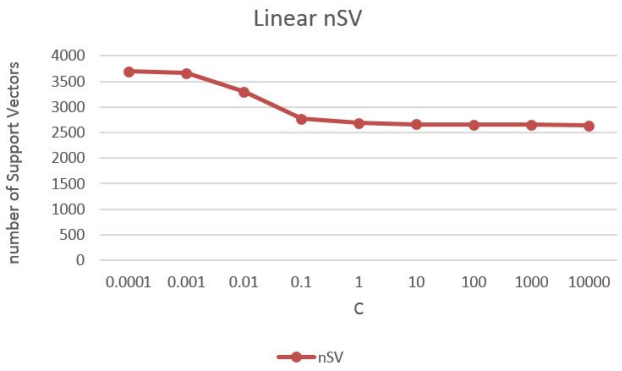



Polynomial kernel d=3
number of Support Vectors
3800
3700
3600
3500
3400
3300
3200
3100
3000
2900
2800
0.0001 0.001 0.01 0.1 1 10 100 1000 10000
C
nSV


Polynomial kernel d=4
Accuracy
1
0.8
0.6
0.4
0.2
0
0.0001 0.001 0.01 0.1 1 10 100 1000 10000
C
TrainAccuracies
ValAccuracies
TestAccuracies


Polynomial kernel d=4
number of Support Vectors
3800
3700
3600
3500
3400
3300
3200
3100
3000
2900
0.0001 0.001 0.01 0.1 1 10 100 1000 10000
C
nSV

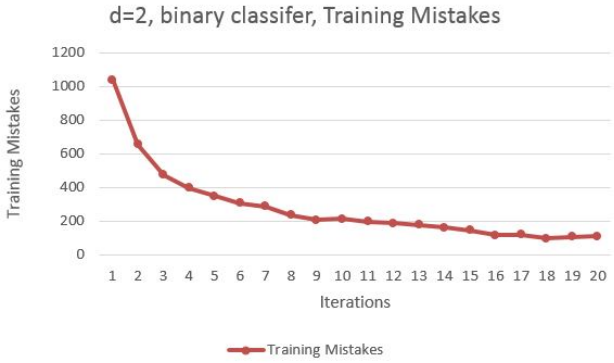

Linear Kernel accuracy comparison
Accuracy
1
0.8
0.6
0.4
0.2
0
0.0001 0.001 0.01 0.1 1 10 100 1000 10000
C
TrainAccuracies
ValAccuracies
TestAccuracies



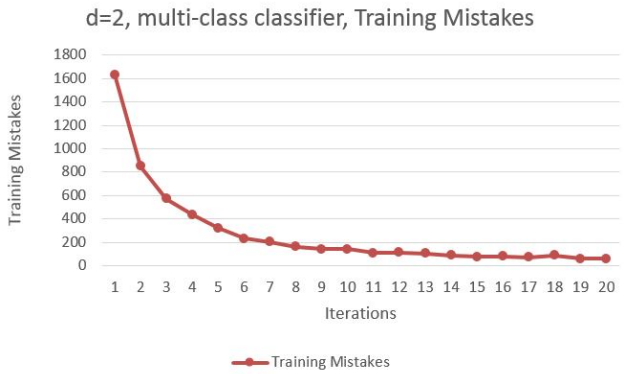
Polynomial kernel performs well when C is large, and also runs much faster than linear model. However, we need to do the comparison test to choose the hyperparameter d.

Problem 10

Answer:
According to the problem 9, choose d =2 that achieves the highest testing accuracy.
a)For binary classifier, number of mistakes as a function of training iterations is shown as:



The corresponding training and testing accuracy is 97.6175005% and 85.5390765%. However, further experiments show that the best testing accuracy for this dataset achieves at $d = 4$, and respectively 99.068% and 89.075%.
b)For multi-class classifier, number of mistakes as a function of training iterations is shown as



The corresponding training and testing accuracy is 98.7221139% and 76.45945%.Further experiments show that it indeed is the best testing accuracy among $d = 2, 3, 4$