Boston University
Department of Electrical and Computer Engineering

ENG EC 414 (Ishwar) Introduction to Machine Learning

HW 7

Issued: Fri 8 Apr 2022 **Due:** 10:55pm Boston time Fri 15 Apr 2022 in Gradescope + Blackboard.
Required reading: Slides on SVM & Multiclass Classification via Binary Classifiers + your lecture notes & Discussion 10.

Important: Before you proceed, please read the documents pertaining to *Homework formatting and submission guidelines* and the *HW-grading policies* in the Homeworks section of Blackboard, especially guidelines for submitting reports in Gradescope. and code in Blackboard. In particular, for computer assignments *you are prohibited from using any online code or built-in MATLAB functions except as indicated in the problem or skeleton code (when provided).*
In order to receive full credit, all work should be supported by a concise explanation that is clear, relevant, specific, logical, and correct. In particular, for each part, you must clearly outline the key steps and provide proper justification for your calculations.

Note: Problem difficulty = number of coffee cups ☕

Problem 7.1 [30pts] Consider two-class classification with the following training feature vectors: $\mathbf{x}_P = (2,0)^\top, \mathbf{x}_Q = (0,4)^\top, \mathbf{x}_R = (3,3)^\top, \mathbf{x}_S = (7,5)^\top$ with labels $-1, -1, +1, +1$ respectively.

(a) [2pts] Hand-plot the training set. Proper labeling of axes and key points is needed to receive full credit.

(b) [1pt] Hand-sketch the hyperplane with parameters $\mathbf{w} = (3,0)^\top$, $b = -3$. Proper labeling of axes and key points is needed to receive full credit.

(c) [2pts] Hand-compute the ℓ_2 distance of $\mathbf{x}_P$ from the hyperplane in part (b).

(d) [3pts] Hand-compute the parameters of the hyperplane in part (b) in *canonical* form with respect to the training set.

(e) [2pts] Determine if the hyperplane in part (b) linearly separates the training set. Explanation is needed to receive credit.

(f) [3pts] Hand-compute the parameters of the hyperplane passing through $\mathbf{x}_P$ and $\mathbf{x}_Q$.

(g) [5pts] Hand-compute the orthogonal projection of $\mathbf{x}_R$ onto the hyperplane in part (f).

(h) [6pts] Hand-compute the parameters of the maximum margin linearly separating hyperplane (SVM hyperplane) in *canonical* form.

(i) [3pts] Determine which training points lie on the marginal SVM hyperplanes. ← $w^T x_j + b = 0$

(j) [3pts] Let $\mathbf{w}_{\text{SVM}} = -\alpha_P \mathbf{x}_P - \alpha_Q \mathbf{x}_Q + \alpha_R \mathbf{x}_R + \alpha_S \mathbf{x}_S$ where the α's are all non-negative. Hand-compute the values of the alphas. Is the solution for α's unique?

Problem 7.2 [102 pts] *(Soft-Margin Binary SVM and All-Pairs Method)*
In this problem, we will implement and use the Stochastic Sub-Gradient Descent (SSGD) Algorithm to learn the parameters of the soft-margin binary SVM classifier. We will use the 3-class *Iris* dataset from Homework 6, but only consider two features, specifically x_2 and x_4, for training and testing all classifiers. We will also implement and test a 3-class classifier using 3 soft-margin binary SVMs and the *All-Pairs* methodology.
Summary of cost function and algorithm for binary soft-margin SVM: Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be a training set of n examples consisting of feature vector $\mathbf{x}_j \in \mathbb{R}^d$ and label $y_j \in \{-1, +1\}$ for each j. Let $\mathbf{x}_j^{\text{ext}} = \begin{pmatrix} \mathbf{x}_j \\ 1 \end{pmatrix}_{(d+1)\times 1}$ denote the extended representation of the j-th feature vector and let $\boldsymbol{\theta} = \begin{pmatrix} \mathbf{w} \\ b \end{pmatrix}$ be the $(d+1) \times 1$ vector of parameters of the soft-margin binary SVM classifier. The cost function (in the unconstrained form) of the binary soft-margin SVM optimization problem for the scaled cumulative slack penalty is given by:

$$g(\boldsymbol{\theta}) = f_0(\boldsymbol{\theta}) + \sum_{j=1}^{n} f_j(\boldsymbol{\theta})$$

where,

$$f_0(\boldsymbol{\theta}) = \frac{1}{2}\|\mathbf{w}\|^2, \quad \text{and} \quad f_j(\boldsymbol{\theta}) = C \max(0, 1 - y_j \boldsymbol{\theta}^\top \mathbf{x}_j^{\text{ext}}), \quad j = 1, \ldots, n,$$

and their sub gradients are given by

$$\nabla_{\boldsymbol{\theta}} f_0(\boldsymbol{\theta}) = \begin{pmatrix} \mathbf{w} \\ 0 \end{pmatrix} = \begin{bmatrix} I_d & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{\theta}, \text{ and } \nabla_{\boldsymbol{\theta}} f_j(\boldsymbol{\theta}) = \begin{cases} -C y_j \mathbf{x}_j^{\text{ext}} & \text{if } y_j \boldsymbol{\theta}^\top \mathbf{x}_j^{\text{ext}} < 1 \\ \mathbf{0} & \text{else.} \end{cases}$$

The pseudocode for implemeting SSGD for soft-margin binary SVM loss is as follows:

Stochastic Sub-Gradient Descent for Soft-Margin Binary SVM

input: Training set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$
initialize: $\boldsymbol{\theta} = \mathbf{0}$
for $t = 1, 2, \ldots, t_{\max}$
  choose sample index: j uniformly at random from $\{1, \ldots, n\}$
  compute sub-gradient:

$$\mathbf{v} = \begin{bmatrix} I_d & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{\theta}$$

  if $(y_j \boldsymbol{\theta}^\top \mathbf{x}_j^{\text{ext}} < 1)$ **then**

$$\mathbf{v} \longleftarrow \mathbf{v} - n\, C y_j\, \mathbf{x}_j^{\text{ext}}$$

  end if
  update parameters:

$$\boldsymbol{\theta} \longleftarrow \boldsymbol{\theta} - s_t \mathbf{v}$$

end for
output: $\boldsymbol{\theta}$

Use the following choices of hyper parameters in all computer experiments for this problem:

$$t_{\max} = 2 \times 10^5, \quad s_t = \frac{0.5}{t}, \quad C = 1.2 \text{ for training all binary classifiers.}$$

We will train a soft-margin binary SVM classfier for every pair of classes and then combine their decisions using the *All-Pairs* methodology.

(a) [5×3 pts] For each binary classifier, create a plot which shows how the sample-normalized cost $\frac{1}{n}g(\boldsymbol{\theta})$ evolves with iteration number t, for every 1000 iterations, i.e., $t = 1000t', t' = 0, 1, 2, 3, \ldots, 200$. Discuss the behavior in terms of both short-term fluctuations and the long-term trend.

(b) [5×3 pts] For each binary classifier, create a plot which shows how the *training* CCR evolves with iteration number, for every 1000 iterations. Discuss the behavior in terms of both short-term fluctuations and the long-term trend. Recall that for a binary SVM

$$\hat{y}_j = \operatorname{sign}\left(\boldsymbol{\theta}^\top \mathbf{x}_j^{\text{ext}}\right), \quad \text{CCR} = \frac{1}{n}\sum_{j=1}^{n} 1(y_j = \hat{y}_j).$$

(c) [5×3 pts] For each binary classifier, create a plot which shows how the *test* CCR evolves with iteration number, for every 1000 iterations. Discuss the behavior in terms of both short-term fluctuations and the long-term trend.

(d) [45pts] *Final values*: After the SSGD algorithm terminates, report the final values of the following for each binary classifier: (i)[3 × 3 pts] $\boldsymbol{\theta}$, (ii)[1 × 3 pts] the training CCR, (iii)[1 × 3 pts] the test CCR, (iv)[5 × 3 pts] training confusion matrix, (v)[5 × 3 pts] test confusion matrix. Discuss your observations.

(e) [12pts] Implement a 3-class classifier using the *All-Pairs* methodology and the 3 binary classifiers corresponding to the final $\boldsymbol{\theta}$ values reported in part (d). Report the following: (i)[1pt] the training CCR, (ii)[1pt] the test CCR, (iii)[5pts] training confusion matrix, (iv)[5pts] test confusion matrix. Discuss your observations.

Code-submission via Blackboard: Create one dot-m file. Name it as follows: `<yourBUemailID>_hw7_2.m` for Problem 7.2. All local functions and scripts pertaining to one problem should appear within the single dot-m file for that problem. When run, your scripts should be able to display in the command window whatever you are asked to compute and report in each part. There is no skeleton code provided for this homework. Create your own code. Reach-out to the TA via Piazza and office/discussion hours for questions related to coding. When submitting code, please include a 'Readme' text file in your source code directory describing approximate running times of different parts, any additional comments (as needed) explaining how to use your code, and any dependencies between different parts. Please do not include into the directory, any data files that are already provided. Write your code under the assumption that all data files are in the same directory as your source code.

Problem 7.1 [30pts] Consider two-class classification with the following training feature vectors: $\mathbf{x}_P = (2,0)^\top, \mathbf{x}_Q = (0,4)^\top, \mathbf{x}_R = (3,3)^\top, \mathbf{x}_S = (7,5)^\top$ with labels $-1,-1,+1,+1$ respectively.

(a) [2pts] Hand-plot the training set. Proper labeling of axes and key points is needed to receive full credit.

(b) [1pt] Hand-sketch the hyperplane with parameters $\mathbf{w} = (3,0)^\top$, $b = -3$. Proper labeling of axes and key points is needed to receive full credit.

(c) [2pts] Hand-compute the ℓ_2 distance of $\mathbf{x}_P$ from the hyperplane in part (b).

(d) [3pts] Hand-compute the parameters of the hyperplane in part (b) in *canonical* form with respect to the training set.

(e) [2pts] Determine if the hyperplane in part (b) linearly separates the training set. Explanation is needed to receive credit.

(f) [3pts] Hand-compute the parameters of the hyperplane passing through $\mathbf{x}_P$ and $\mathbf{x}_Q$.

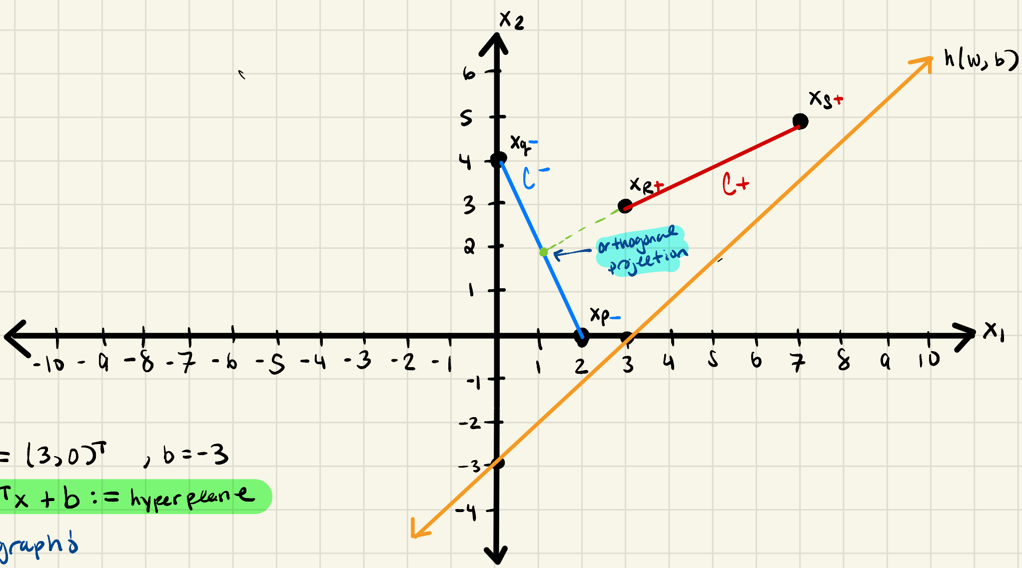
(g) [5pts] Hand-compute the orthogonal projection of $\mathbf{x}_R$ onto the hyperplane in part (f).

(h) [6pts] Hand-compute the parameters of the maximum margin linearly separating hyperplane (SVM hyperplane) in *canonical* form.

(i) [3pts] Determine which training points lie on the marginal SVM hyperplanes. ← x_R & x_S

(j) [3pts] Let $\mathbf{w}_{\text{SVM}} = -\alpha_P \mathbf{x}_P - \alpha_Q \mathbf{x}_Q + \alpha_R \mathbf{x}_R + \alpha_S \mathbf{x}_S$ where the α's are all non-negative. Hand-compute the values of the alphas. Is the solution for α's unique?

a)



b) $w = (3,0)^T$, $b = -3$

$w^T x + b :=$ hyperplane

On graph ↑

c) $\text{dist}(x_P, hp(w,b)) = \dfrac{|w^T x_P + b|}{\|w\|_2}$ ← not generic but equation needed

$\|w\| = \sqrt{3^2} = 3$

$\boxed{\text{dist}(x_P, hp(w,b)) = 1}$

$\text{dist}(x_P, hp(w,b)) = \dfrac{|3(2) - 3|}{3} = \dfrac{3}{3} = 1$

d)

- Then $\gamma = \min_{j=1,\dots,n} \left|\mathbf{w}^{\mathrm{T}} \mathbf{x}_j + b\right| > 0$
- $\Rightarrow \min_{j=1,\dots,n} \left|\left(\frac{\mathbf{w}}{\gamma}\right)^{\mathrm{T}} \mathbf{x}_j + \left(\frac{b}{\gamma}\right)\right| = 1$

x_p: $w^T x_p + b = 3$ ← min where >0

x_q: $w^T x_q + b = -3$

$$\begin{vmatrix} 3 & 0 \end{vmatrix} \begin{vmatrix} 0 \\ 4 \end{vmatrix} - 3 = 0 + 0 - 3 = -3$$

x_R: $w^T x_R + b = 6$

$$\begin{vmatrix} 3 & 0 \end{vmatrix} \begin{vmatrix} 3 \\ 3 \end{vmatrix} - 3 = 9 - 3 = 6$$

x_S: $w^T x_S + b = 18$

$$\begin{vmatrix} 3 & 0 \end{vmatrix} \begin{vmatrix} 7 \\ 5 \end{vmatrix} - 3 = 21 - 3 = 18$$

Only one column in x_p

$$\begin{vmatrix} 3 & 0 \end{vmatrix} \begin{vmatrix} 2 \\ 0 \end{vmatrix} - 3 = 1$$

$$\frac{6}{3} - \frac{3}{3} = 1$$

2 -1

$\gamma = 3$

$\boxed{\left((1,0)^T, -1 \right)}$

e)

Geometric Margin of $hp(\mathbf{w}, b)$ = distance to closest $\mathbf{x}_j$ in $\mathcal{D}$:

$$\rho(\mathbf{w}, b, \mathcal{D}) = \min_{j=1,\ldots,n} \frac{|\mathbf{w}^{\mathrm{T}} \mathbf{x}_j + b|}{||\mathbf{w}||}$$

$x_p =$ dist$(x_p, h(w,b)) = 1$

$x_q = \frac{|w^T x_q + b|}{||w||} = 1$ minimum

$\frac{|-3|}{3} = 1$

$x_R: \frac{|w^T x_R + b|}{||w||} = 2$

$\frac{|b|}{3} = 2$

$x_S: \frac{|w^T x_S + b|}{||w||} = 6$

$\rho(w, b, D) = 1$

The hyperplane in part b does not linearly seperate the dataset because it does not classify all training examples without error.

f) $\begin{vmatrix}2 & 0\\ 0 & 4\end{vmatrix}\begin{vmatrix}w_1\\ w_2\end{vmatrix} = -b\begin{vmatrix}1\\ 1\end{vmatrix}$ $\quad Ax = -b$, $b = 1$

$\frac{1}{8}\begin{vmatrix}4 & 0\\ 0 & 2\end{vmatrix} = \begin{vmatrix}1/2 & 0\\ 0 & 1/4\end{vmatrix}$

$X = A^{-1}B$

$\begin{vmatrix}1/2 & 0\\ 0 & 1/4\end{vmatrix}\begin{vmatrix}-b\\ -b\end{vmatrix} = \begin{vmatrix}-1/2b\\ -1/4b\end{vmatrix}$

$2\times 2 \qquad 2\times 1$

$\boxed{hp(b(-1/2, -1/4)^T, b)}$

g) $proj_{hp(w,b)}(x) = x - \frac{(w^Tx + b)}{\|w\|_2}\cdot\frac{w}{\|w\|_2}$

$proj_{hp(w,b)}(x_R) = x_R - \frac{(w^Tx_R + b)}{\|w\|_2}\cdot\frac{w}{\|w\|_2}$

x_R: $w^Tx_R + b$

$|-b/2 \;\; -b/4|\begin{vmatrix}3\\ 3\end{vmatrix} + b = 3\left(-\frac{b}{2}\right) + 3\left(-\frac{b}{4}\right) + b = -5b/4$

$\|w\| = \sqrt{(-b/2)^2 + (-b/4)^2} = \sqrt{b^2/4 + b^2/16} = \sqrt{5b^2/16}$

$proj_{hp(w,b)}(x_R) = x_R - \frac{-5b/4}{\sqrt{5b^2/16}}\cdot\frac{(-b/2, -b/4)}{\sqrt{5b^2/16}}$

$= x_R - \frac{-5b/4}{5b^2/16}\cdot(-b/2, -b/4)$

$= x_R + \frac{4}{b}(-b/2, -b/4)$

$= (3-2, 3-1)$

closest point $\rightarrow$ $\boxed{proj_{hp(w,b)}(x_R) = (1, 2)}$

h) $w_{svm} = \frac{2(x_+ - x_-)}{\|x_+ - x_-\|^2}$ $\qquad b_{svm} = -w_{svm}^T\frac{(x_+ + x_-)}{2}$

$(x_+ - x_-) = (3,3) - (1, 2)$
$= (3-1, 3-2) = (2, 1)$

$2(x_+ - x_-) = (4, 2)$

$\|x_+ - x_-\| = \sqrt{(4)^2 + (2)^2}$
$= \sqrt{16 + 4} = \sqrt{20}$

$\|x_+ - x_-\|^2 = (\sqrt{20})^2 = 20$

$w_{svm} = \frac{(4,2)}{20} = (4/20, 2/20)$

$\boxed{w_{svm}^T = (1/5, 1/10)}$

$(x_+ + x_-) = (3,3) + (1,2)$
$= (4, 5)$

$\frac{(x_+ + x_-)}{2} = (2, 5/2)$

$b_{svm} = -(1/5, 1/10)\cdot\begin{vmatrix}2\\ 5/2\end{vmatrix}$

$b_{svm} = -2/5 - 1/10(5/2)$
$= -13/20$

$\boxed{b_{svm} = -13/20}$

$\gamma = \frac{2}{\|x_+ - x_-\|^2}$

$$\left(\frac{1/5}{\gamma}, \frac{1/10}{\gamma}\right)\begin{vmatrix}3\\3\end{vmatrix} \quad \frac{-13/20}{\gamma} = 1$$

$$\frac{(1/5)(3) + (1/10)(3) - 13/20}{\gamma} = 1$$

$$\frac{3/5 + 3/10 - 13/20}{\gamma} = 1$$

$$\frac{1}{4\gamma} = \frac{1}{1} \rightarrow \gamma = 1/4$$

$$\boxed{(4/5, 4/10)^T, -13/5)}$$

$$\begin{vmatrix}4/5\\4/10\end{vmatrix} = -\alpha_p\begin{vmatrix}2\\0\end{vmatrix} - \alpha_q\begin{vmatrix}0\\4\end{vmatrix} + \tfrac{2}{3}\alpha_R\begin{vmatrix}3\\3\end{vmatrix} + \tfrac{2}{5}\alpha_S\begin{vmatrix}7\\5\end{vmatrix}$$

$\frac{2}{3}$ $\frac{14}{5}$

$x + 4 = \frac{14}{5}$

$x = -\frac{6}{5}$

$x - 4 = 4/10$

$x = -4/10 + 4 = \frac{22}{5}$ $\quad 2 \quad \frac{10}{5}$

$x = 4/5 + 2 = \frac{14}{5}$

$\alpha_p = 1$, $\alpha_q = -\frac{6}{5}$, $\alpha_R = 2/3$, $\alpha = \frac{2}{5}$

No since w_{svm} is a linear combination the solution for α's is unique.

i) By inspection of graph, x_R & x_S most likely lie on the hyperplane.

x_R: $w^T_{svm} x_R + b_{svm} =$

$3(1/5) + 3(1/10) - 13/20 =$

$3/5 + 3/10 - 13/20 =$

$12/20 + 6/20 - 13/20 = 5/20$

x_S: $w^T x_S + b_{svm}$

$7(1/5) + 5(1/10) - 13/20 =$

$7/5 + 5/10 - 13/20 =$

$28/20 + 10/20 - 13/20 = 25/20$

No points lie on the hyperplane

j) $w_{svm} = -\alpha_p x_p - \alpha_q x_q + \alpha_R x_R + \alpha_S x_S$

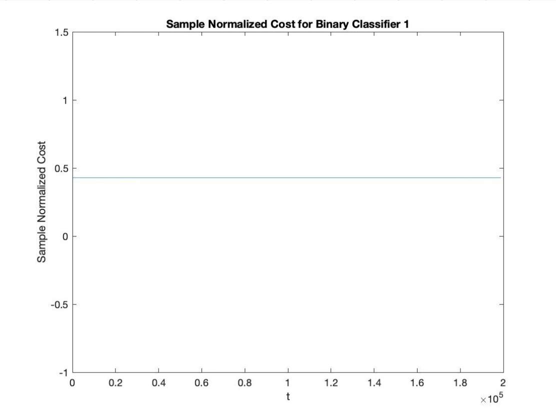
$$\underline{w}_{svm} = \sum_{j=1}^{n} \alpha_j \underline{x}_j$$

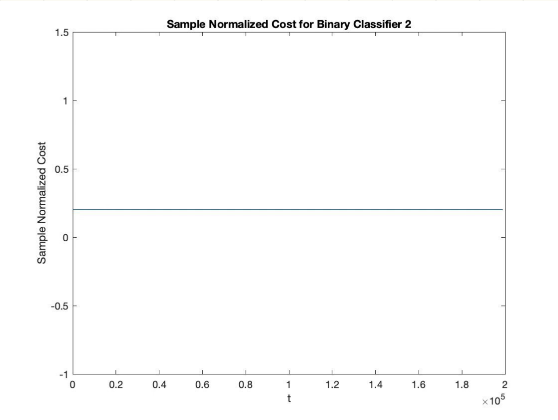
$w_{svm} = \gamma(x_+ - x_-) \leftarrow$ direction

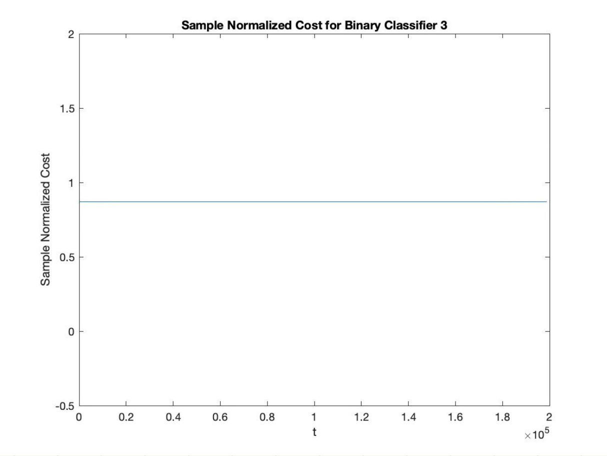
write x_+ as a linear combo, then express x_- as a linear combo. what's linear combo?

7.2

(a) [5×3 pts] For each binary classifier, create a plot which shows how the sample-normalized cost $\frac{1}{n}g(\boldsymbol{\theta})$ evolves with iteration number t, for every 1000 iterations, i.e., $t = 1000t', t' = 0, 1, 2, 3, \ldots, 200$. Discuss the behavior in terms of both short-term fluctuations and the long-term trend.



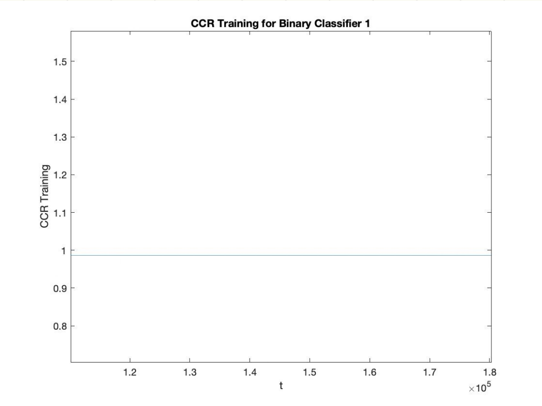


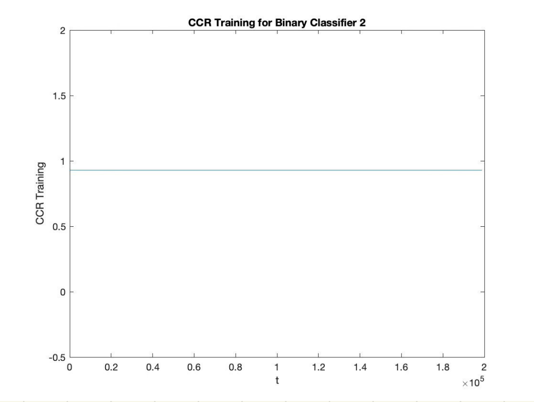


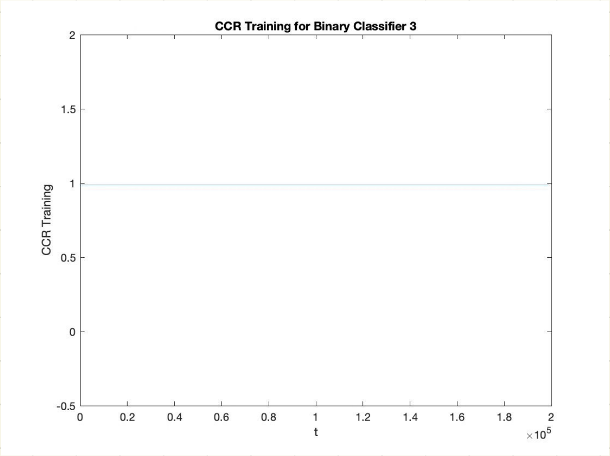
Normalized cost is a constant value. The value fluctuates between each classifier in a range of [0, 1].

(b) [5×3 pts] For each binary classifier, create a plot which shows how the *training* CCR evolves with iteration number, for every 1000 iterations. Discuss the behavior in terms of both short-term fluctuations and the long-term trend. Recall that for a binary SVM

$$\hat{y}_j = \text{sign}\left(\boldsymbol{\theta}^\top \mathbf{x}_j^{\text{ext}}\right), \quad \text{CCR} = \frac{1}{n}\sum_{j=1}^{n} 1(y_j = \hat{y}_j).$$

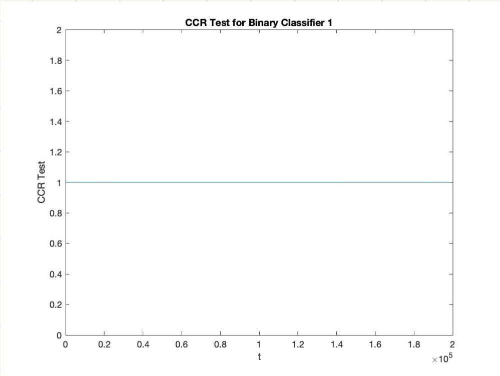


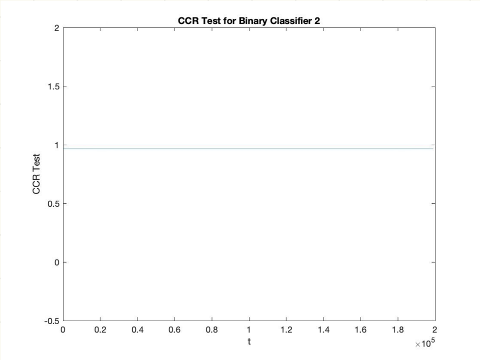


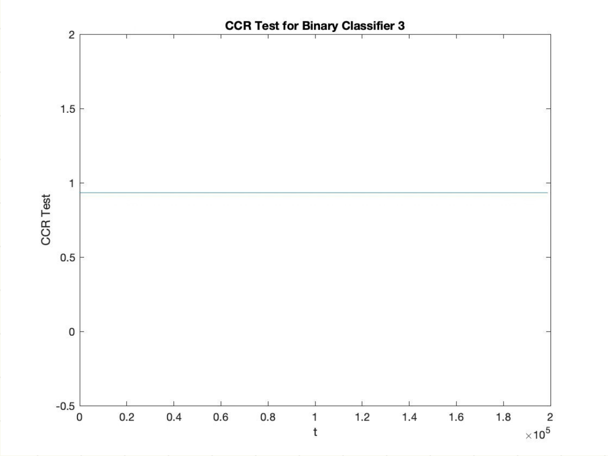


For all 3 of the classifiers, the CCR training stays at a constant value of about 98%.

(c) [5×3 pts] For each binary classifier, create a plot which shows how the *test* CCR evolves with iteration number, for every 1000 iterations. Discuss the behavior in terms of both short-term fluctuations and the long-term trend.







For all 3 of the classifiers, the CCR test stays at a constant value of 1. The training data successful predicted the values of iris dataset features 2 & 4.

(d) [45pts] *Final values*: After the SSGD algorithm terminates, report the final values of the following for each binary classifier: (i)[3×3 pts] $\boldsymbol{\theta}$, (ii)[1×3 pts] the training CCR, (iii)[1×3 pts] the test CCR, (iv)[5×3 pts] training confusion matrix, (v)[5×3 pts] test confusion matrix. Discuss your observations.

i)

```
Final Theta for Binary Classifier 1
    2.8419
   -2.2071
   -6.8723

Final Theta for Binary Classifier 2
    0.7963
   -3.5957
    3.6697

Final Theta for Binary Classifier 3
    3.9088
   -1.9714
  -10.1354
```

ii)

```
Final training CCR for Binary Classifier 1
    0.9857

Final training CCR for Binary Classifier 2
    0.9857

Final training CCR for Binary Classifier 3
    0.9857
```

iii)

```
Final test CCR for Binary Classifier 1
    1

Final test CCR for Binary Classifier 2
    1

Final test CCR for Binary Classifier 3
    1
```

iv)

```
Training Confusion matrix for Binary Classifier 1
    35     0
     1    34

Training Confusion matrix for Binary Classifier 2
    32     3
     2    33

Training Confusion matrix for Binary Classifier 3
    35     0
     1    34
```

v)

```
Test Confusion matrix for Binary Classifier 1
    15     0
     0    15

Test Confusion matrix for Binary Classifier 2
    14     1
     0    15

Test Confusion matrix for Binary Classifier 3
    13     2
     0    15
```

The corresponding thetas contributed to a high CCR rate for both the training and test data.

(e) [12pts] Implement a 3-class classifier using the *All-Pairs* methodology and the 3 binary classifiers corresponding to the final $\boldsymbol{\theta}$ values reported in part (d). Report the following: (i)[1pt] the training CCR, (ii)[1pt] the test CCR, (iii)[5pts] training confusion matrix, (iv)[5pts] test confusion matrix. Discuss your observations.

i)

```
Final training CCR (All Pairs Method) for Binary Classifier 1
   0.5000

Final training CCR (All Pairs Method) for Binary Classifier 2
   0.5000

Final training CCR (All Pairs Method) for Binary Classifier 3
   0.5000
```

ii)

```
Final test CCR (All Pairs Method) for Binary Classifier 1
   1

Final test CCR (All Pairs Method) for Binary Classifier 2
   1

Final test CCR (All Pairs Method) for Binary Classifier 3
   1
```

iii)

Training Confusion matrix (All Pairs Method) for Binary Classifier 1

0	0	0	0
0	35	0	0
3	0	0	0
32	0	0	0

Training Confusion matrix (All Pairs Method) for Binary Classifier 2

0	0	0	0
2	33	0	0
0	3	0	0
32	0	0	0

Training Confusion matrix (All Pairs Method) for Binary Classifier 3

0	0	0	0
0	35	0	0
2	1	0	0
32	0	0	0

iv)

Test Confusion matrix (All Pairs Method) for Binary Classifier 1

0	0	0	0
0	15	0	0
1	0	0	0
14	0	0	0

Test Confusion matrix (All Pairs Method) for Binary Classifier 2

0	0	0	0
0	15	0	0
0	1	0	0
14	0	0	0

Test Confusion matrix (All Pairs Method) for Binary Classifier 3

0	0	0	0
0	15	0	0
0	1	0	0
14	0	0	0

The All Pairs method did not work as well as SVM in training