Cover Sheet

Assignment Submission Fill in and include this cover sheet with each of your assignments. Assignments are due at 11:59pm. All students (SCPD and non-SCPD) must submit their homeworks via GradeScope (`http://www.gradescope.com`). Students can typeset or scan their homeworks. Make sure that you answer each question on a separate page. Students also need to upload their code at `http://snap.stanford.edu/submit`. Put all the code for a single question into a single file and upload it. Please do not put any code in your GradeScope submissions.

Late Day Policy Each student will have a total of *two* free late periods. *One late period expires at the start of each class.* (Homeworks are usually due on Thursdays, which means the first late periods expires on the following Tuesday.) Once these late periods are exhausted, any assignments turned in late will be penalized 50% per late period. However, no assignment will be accepted more than *one* late period after its due date.

Honor Code We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down their solutions independently i.e., each student must understand the solution well enough in order to reconstruct it by him/herself. Students should clearly mention the names of all the other students who were part of their discussion group. Using code or solutions obtained from the web (github/google/previous year solutions etc.) is considered an honor code violation. We check all the submissions for plagiarism. We take the honor code very seriously and expect students to do the same.

Your name: MANOJ RAJ
Email: manoj1390@gmail.com **SUID:** manojr

Discussion Group: ______

I acknowledge and accept the Honor Code.

(Signed) Manoj

CS246: Mining Massive Datasets
Homework 4

Answer to Question 1(a)

w	w_1	w_2	w_3
Initial	0	0	0
after observing PID = 1	0	0	-0.2
PID = 2	0	0	-0.2
PID = 3	0	0	-0.2
PID = 4	0	-0.2	0

1st : $x \cdot w = 0 \cdot 0 + 0 \cdot 0 + 0 \cdot (-1) = 0$ (misclassified)

$w' = (0,0,0) + 0.2(+1)(0,0,-1) = (0,0,0) + (0,0,-0.2) = (0,0,-0.2)$

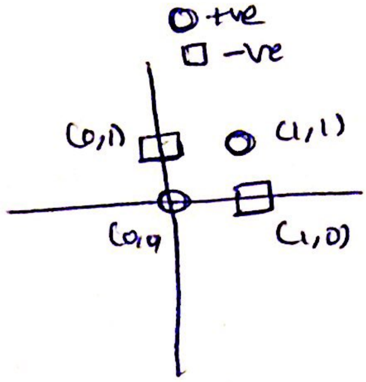
2nd $(1,1,-1)$ $x \cdot w' = 1*0 + 1*0 + (-1)(-0.2) = 0+0+0.2 = 0.2 > 0$ $+1$ $(\checkmark)$

3rd $(1,1,-1)$ $x \cdot w' = 1*0 + 1*0 + (-1)(-0.2) = 0.2$ > 0 $+1$ $(\checkmark)$

4th $(0,1,-1)$ $x \cdot w' = 0*0 + 1*0 + (-1)(-0.2) = 0.2$ $> +1$ $(\times)$ (misclassified)

$w_2' = (0,0,-0.2) + 0.2(-1)(0,1,-1) = (0,0,-0.2) + (0,-0.2,0.2)$
$= (0,-0.2,0)$

Answer to Question 1(b)



Excluding the bias we can see that in a two dimensional plane the given set of ~~bou~~ data points are NOT linearly separable because they are configured such that there is no plane $h \in R^2$ and $\beta \in R$ with which we could say $a \in +ve_y : h^T a > \beta$ and for all $b \in -ve_y : h^T b < \beta$. Having the bias also just a linear change. Hence, there will be no change to linear separability. Hence the perceptron algorithm cannot return a solution which is linearly separable with existing features.

(iii) $\Psi = \{\phi_1 - \phi_2, \phi_1 + \phi_2 - 1)$

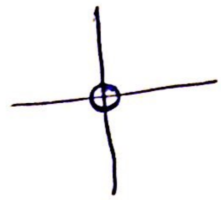
This is just a linear change to features (rotates/scales) so, still is not separable

Answer to Question 1(c)

$\Psi = (\phi_1^2, \phi_1 \phi_2, -1)$

(i)

	ϕ_1	ϕ_2	ϕ_3	Label	ϕ_1'	ϕ_2'	ϕ_3'
1	0	0	-1	+1	0	0	-1
	1	1	-1	+1	1	1	-1
	1	1	-1	+1	1	1	-1
-4	0	1	-1	-1	0	0	-1
	1	0	-1	-1	1	0	-1



Not linearly separable since 1 & 4 with opp polarities map to same pts

$\Psi = ((\phi_1 \text{ xor } \phi_2), \phi_2, -1)$

(ii)

ϕ_1	ϕ_2	ϕ_3	Label	ϕ_1'	ϕ_2'	ϕ_3'
0	0	-1	+1	0	0	-1
1	1	-1	+1	0	1	-1
1	1	-1	+1	0	1	-1
0	1	-1	-1	1	1	-1
1	0	-1	-1	1	0	-1

Yes this is linearly separable 1st 3 points map to $(0,0,-1)$ and $(0,1,-1)$ and last two point to $(1,1,-1)$ and $(1,0,-1)$

$w_1 * 0 + w_2 * 0 + w_3 * (-1) \geq 0$; $w_1 * 0 + w_2 * 1 - w_3 * 1 \geq 0$

$\boxed{w_3 \leq 0}$ $\boxed{w_2 - w_3 \geq 0}$

$\boxed{w_2 \geq w_3}$

$w_1 + w_2 - w_3 < 0$

$w_1 - w_3 < 0$

$\boxed{w_1 < w_3}$

$\boxed{w_1 + w_2 < w_3}$

$-2, 0, -1$

$w = [-2, 0, -1]$ separates two classes.

Answer to Question 2(a)

$$\nabla_b f(w,b) = \frac{\partial f}{\partial b}(w,b) = C \sum_{i=1}^{n} \frac{\partial \xi}{\partial b} L(x_i, y_i)$$

$$\frac{\partial \xi}{\partial b} L(x_i, y_i) = \begin{cases} 0 & \text{if } y_i(x_i \cdot w + b) \geq 1 \\ -y_i x_i(j) & \text{otherwise} \end{cases}$$

Answer to Question 2(b)

Batch gradient : 57 iterations -> 21.68 s (0.38 s / iteration)

Stochastic gradient : 497 iterations -> 155.18 s (0.31 s / iteration)

Minibatch gradient : 746 iterations -> 226.57 s (0.303 / iteration)

Batch gradient converged at a higher cost function than mini-batch. It converged very quickly and reduced the cost monotonously in every iteration

~~Despite taking more time~~,

The time for iteration for batch gradient is the most highest. But at the point of its convergence other methods still have ~~the~~ high cost values.

Answer to Question 3(a)

$$I(D) = 100 * \left[1 - \left(\frac{60}{100}\right)^2 - \left(\frac{40}{100}\right)^2\right] = 48$$

for "likes wine" $|D_L| = |D_R| = 50$,

$$I(D_L) = I(D_R) = 50 * \left[1 - \left(\frac{30}{50}\right)^2 - \left(\frac{20}{50}\right)^2\right] = 24$$

$$I(D) - I(D_L) - I(D_R) = 0$$

For "likes running" $|D_L| = 30$, $|D_R| = 70$

$$I(D_L) = 30 * \left[1 - \left(\frac{20}{30}\right)^2 - \left(\frac{10}{30}\right)^2\right] = 13.33$$

$$I(D_R) = 70 * \left[1 - \left(\frac{40}{70}\right)^2 - \left(\frac{30}{70}\right)^2\right] = 34.29$$

$$I(D) - I(D_L) - I(D_R) = 0.38$$

For "likes pizza" $|D_L| = 80$, $|D_R| = 20$

$$I(D_L) = 80 * \left[1 - \left(\frac{50}{80}\right)^2 - \left(\frac{30}{80}\right)^2\right] = 37.5$$

$$I(D_R) = 20 * \left[1 - \left(\frac{10}{20}\right)^2 - \left(\frac{10}{20}\right)^2\right] = 10$$

$$I(D) - I(D_L) - I(D_R) = 0.5$$

We will use the "likes pizza" binary value since it has the highest value of gini index G.

Answer to Question 3(b)

The decision tree identifies a_1 as the most important attribute and ~~other~~ will have it at its root. The other attributes would be in the rest of the parts of the tree.

The above is more likely to overfit the tree.

The desired decision tree would contain just 1 split on a_1 with + label if $a_1 = 1$ and − label if $a_1 = 0$ so that the model will avoid overfitting & predict with highest accuracy

Given: $\tilde{F}[i] \geq F[i]$

Answer to Question 4(a)

$E[c_{j,h_j(i)}] \leq F[i] + \frac{\epsilon}{e}(t - F[i])$

To prove: $Pr[\tilde{F}[i] \leq F[i] + \epsilon t] \geq 1-\delta$

$\downarrow$ LHS

$1 - Pr[\tilde{F}[i] > F[i] + \epsilon t]$

For every item in data stream i $c_{j,h_j(i)}$ $(\forall 1 \leq j \leq \lceil \log(\frac{1}{\delta}) \rceil)$ will increase by 1

a word x occurs $F[x]$ times. Also there is a chance of other items getting hashed to $c_{j,h_j(x)}$

Thus $c_{j,h_j(x)} \geq F[x]$

$\therefore$ $Pr[\tilde{F}[i] > F[i] + \epsilon t] \leq Pr[c_{j,h_j(i)} > F[i] + \epsilon t]$

By independence of hash functions,

$$Pr[c_{j,h_j(i)} > F[i] + \epsilon t], \forall 1 \leq j \leq \lceil \log(1/\delta) \rceil = \prod_{j=1}^{\lceil \log(1/\delta) \rceil} Pr[c_{j,h_j(i)} > F[i] + \epsilon t]$$

By Markov's inequality,

$$Pr[c_{j,h_j(i)} > F[i] + \epsilon t] \leq \frac{E[c_{j,h_j(i)}] - F[i]}{\epsilon t} \leq \frac{t - F[i]}{e t} \leq \frac{1}{e}$$

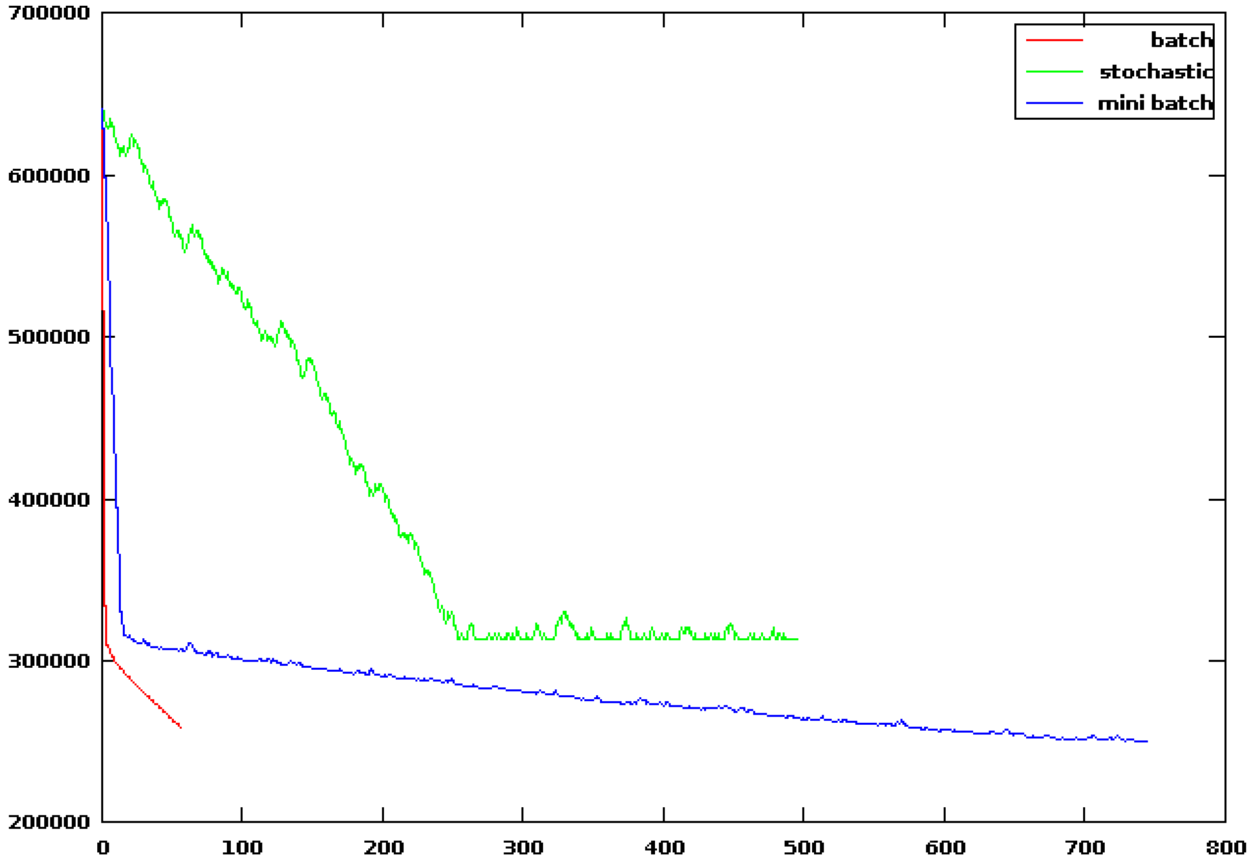
$$\Rightarrow Pr[\tilde{F}[i] > F[i] + \epsilon t] \leq \left(\frac{1}{e}\right)^{\lceil \log(1/\delta) \rceil} \leq \frac{1}{e}^{\log(1/\delta)} \Big\} = \delta$$

$\therefore$ $Pr[\tilde{F}[i] \leq F[i] + \epsilon t] \geq 1 - \delta$

Answer to Question 4(b)

For relative frequencies larger than 10^{-6}, the relative error falls below 1 (10^{0})

Question 2 Graph:



Question 4: Graph