CSE 250A Hwk 8 Andrew Ghafari A59020215.

Question 1:

a) Python

b) $P\left(\{R_j = r_j^{(t)}\}_{j \in \Omega_t}\right)$

$= \sum_{i=1}^{k} P\left(\{R_j = r_j^{(t)}\}_{j \in \Omega_t}, Z=i\right)$ → marginalization

$= \sum_{i=1}^{k} P(Z=i) \cdot P\left(\{R_j = r_j^{(t)}\}_{j \in \Omega_t} \mid Z=i\right)$ Product Rule

$= \sum_{i=1}^{k} P(Z=i) \cdot \prod_{j \in \Omega_t} P\left(\{R_j = r_j^{(t)} \mid Z=i\right)$ CI dii

c) $P\left(Z=i \mid \{R_j = r_j^{(t)}\}_{j \in \Omega_t}\right)$

$$= \frac{P\left(Z=i, \{R_j = r_j^{(t)}\}_{j \in \Omega_t}\right)}{P\left(\{R_j = r_j^{(t)}\}_{j \in \Omega_t}\right)}$$

(b) →

$$= \frac{P(Z=i) \cdot P\left(\{R_j = r_j^{(t)}\}_{j \in \Omega_t} \mid Z=i\right)}{\sum_{i=1}^{k} P(Z=i) \cdot \prod_{j \in \Omega_t} P\left(\{R_j = r_j^{(t)} \mid Z=i\right)}$$ ← also solved in (b) using CI.

$$= \frac{P(Z=i) \prod_{j \in W_t} P\left(\{R_j = r_j^{(t)} \mid Z=i\right)}{\sum_{i=1}^{k} P(Z=i) \prod_{j \in W_t} P\left(\{R_j = r_j^{(t)} \mid Z=i\right)}$$

d) <u>M Step:</u>

$$p_{it} = P\left(Z=i \mid \{R_j = r_j^{(t)}\}_{j \in W_t}\right)$$

$$P(Z=i) \leftarrow \frac{1}{T} P\left(Z=i \mid \{R_j = r_j^{(t)}\}_{j \in W_t}\right) = \frac{1}{T} \sum_{t=1}^{T} p_{it}$$

$$P(R_j = 1 \mid Z=i) \leftarrow \frac{① \sum_t P\left(R_j = 1, Z=i \mid \{R_k = r_k^{(t)}\}_{k \in W_t}\right)}{② \sum_t \underbrace{P\left(Z=i \mid \{R_k = r_k^{(t)}\}_{k \in W_t}\right)}_{p_{it}}}$$

$$① = \sum_{\{t \mid j \in W_t\}} I(r_j^{(t)}, 1) \cdot P\left(Z=i \mid \{R_k = r_k^{(t)}\}_{k \in W_t}\right)$$
$$+ \sum_{\{t \mid j \notin W_t\}} P\left(Z=i \mid \{R_k = r_k^{(t)}\}_{k \in W_t}\right) \cdot P\left(R_j = 1 \mid Z=i, \{R_k = r_k^{(t)}\}_{k \in W_t}\right)$$

$$= \sum_{\{t \mid j \in W_t\}} I(r_j^{(t)}, 1) \cdot p_{it} + \sum_{\{t \mid j \notin W_t\}} p_{it} \cdot P(R_j = 1 \mid Z=i)$$

$$② = \sum_{t=1}^{T} p_{it}$$

All together:

$$P(R_j=1|Z=i) \leftarrow \frac{\sum_{\{t|j \in U_t\}} I(r_j^{(t)},1)\,p_{it} + \sum_{\{t|j \notin U_t\}} P(R_j=1|Z=i)\,p_{it}}{\sum_{t=1} p_{it}}$$

f) g) Python.

Question 2:

a) $$P(y=1|\vec{x}) = \frac{P(y=1).P(\vec{x}|y=1)}{P(\vec{x})}$$

$$= \frac{P(y=1).P(\vec{x}|y=1)}{P(y=0).P(\vec{x}|y=0) + P(y=1).P(\vec{x}|y=1)}$$

$$= \frac{\pi_1 . 2\pi^{-d/2} . |\Sigma_1|^{-1/2} e^{-1/2(\vec{x}-\mu_1)^T \Sigma_1^{-1}(\vec{x}-\mu_1)}}{\pi_1 . 2\pi^{-d/2} . |\Sigma_1|^{-1/2} e^{-1/2(\vec{x}-\mu_1)^T \Sigma_1^{-1}(\vec{x}-\mu_1)} + \pi_0 . 2\pi^{-d/2} . |\Sigma_0|^{-1/2} e^{-1/2(\vec{x}-\mu_0)^T \Sigma_0^{-1}(\vec{x}-\mu_0)}}$$

b) $$P(y=1|\vec{x}) = \frac{\pi_1 \cancel{(2\pi)^{-d/2}} . \cancel{|\Sigma|^{-1/2}} . e^{-1/2(\vec{x}-\mu_1)^T \Sigma^{-1}(\vec{x}-\mu_1)}}{\pi_1 (2\pi)^{-d/2} . |\Sigma|^{-1/2} . e^{-1/2(\vec{x}-\mu_1)^T \Sigma^{-1}(\vec{x}-\mu_1)} + \pi_0 \cancel{(2\pi)^{-d/2}} . \cancel{|\Sigma|^{-1/2}} e^{-1/2(\vec{x}-\mu_0)^T \Sigma^{-1}(\vec{x}-\mu_0)}}$$

Scanned with CamScanner

$$= \frac{\pi_1 e^{-\frac{1}{2}(\vec{x}-\mu_1)^T \Sigma^{-1} (\vec{x}-\mu_1)}}{\pi_1 e^{-\frac{1}{2}(\vec{x}-\mu_1)^T \Sigma^{-1} (\vec{x}-\mu_1)} + \pi_0 e^{-\frac{1}{2}(\vec{x}-\mu_0)^T \Sigma^{-1} (\vec{x}-\mu_0)}}$$

if we factorize by the numerator up & down we get.

$$= \frac{1}{1 + \frac{\pi_0}{\pi_1} e^{\underbrace{-\frac{1}{2}(\vec{x}-\mu_0)^T \Sigma^{-1} (\vec{x}-\mu_0) + \frac{1}{2}(\vec{x}-\mu_1)^T \Sigma^{-1} (\vec{x}-\mu_1)}_{(1)}}}$$

$$(1) = \left(-\frac{1}{2}\vec{x}^T + \frac{1}{2}\vec{\mu_0}^T\right)\left(\Sigma^{-1}\vec{x} - \Sigma^{-1}\vec{\mu_0}\right)$$

$$+ \left(\frac{1}{2}\vec{x}^T - \frac{1}{2}\vec{\mu_1}^T\right)\left(\Sigma^{-1}\vec{x} - \Sigma^{-1}\vec{\mu_1}\right)$$

$$= \Sigma^{-1}\Big(\cancel{-\frac{1}{2}\vec{x}^T\vec{x}} + \frac{1}{2}\vec{x}^T\vec{\mu_0} + \frac{1}{2}\vec{\mu_0}^T\vec{x} - \frac{1}{2}\vec{\mu_0}^T\vec{\mu_0}$$

$$\cancel{+\frac{1}{2}\vec{x}^T\vec{x}} - \frac{1}{2}\vec{x}^T\vec{\mu_1} - \frac{1}{2}\vec{\mu_1}^T\vec{x} + \frac{1}{2}\vec{\mu_1}^T\vec{\mu_1}\Big)$$

w/ $A^T B = B^T A$ will get

$$= \Sigma^{-1}\Big(\frac{1}{2}\vec{x}^T\vec{\mu_0} + \frac{1}{2}\vec{x}^T\vec{\mu_0} - \frac{1}{2}\vec{\mu_0}^T\vec{\mu_0}$$

$$-\frac{1}{2}\vec{x}^T\vec{\mu_1} - \frac{1}{2}\vec{x}^T\vec{\mu_1} + \frac{1}{2}\vec{\mu_1}^T\vec{\mu_1}\Big)$$

$$= \Sigma^{-1}\left[\left(\vec{x}^T\vec{\mu_0} - \frac{1}{2}\vec{\mu_0}^T\vec{\mu_0}\right) + \left(\frac{1}{2}\vec{\mu_1}^T\vec{\mu_1} - \vec{x}^T\vec{\mu_1}\right)\right]$$

$$= \Sigma^{-1}\left(\vec{\mu_0}^T\vec{x} - \frac{1}{2}\vec{\mu_0}^T\vec{\mu_0}\right) + \left(\frac{1}{2}\vec{\mu_1}^T\vec{\mu_1}\right) - \vec{\mu_1}^T\vec{x}\Big)$$

$$= \Sigma^{-1}\left(\vec{x}\left(\vec{\mu_0} - \vec{\mu_1}\right)^T\right) + \frac{1}{2}\left(\vec{\mu_1}^T\vec{\mu_1} - \vec{\mu_0}^T\vec{\mu_0}\right)\Big)$$

$$= \alpha$$

$$P(y=1/\vec{x}) = \frac{1}{1 + \frac{\pi_0}{\pi_1}e^{\alpha}} = \frac{1}{1 + e^{\log\frac{\pi_0}{\pi_1}}e^{\alpha}}$$

$$= \frac{1}{1 + e^{\log \frac{\pi_0}{\pi_1} + \alpha}}$$

$$= \frac{1}{1+e^{-z}} \quad \text{where } z = -\left(\log \frac{\pi_0}{\pi_1} + \alpha\right)$$

$$= -\Sigma^{-1}\left[\vec{x}(\vec{\mu_0} - \vec{\mu_1})^T + \frac{1}{2}\left(\vec{\mu_1}^T \vec{\mu_1} - \vec{\mu_0}^T \vec{\mu_0}\right)\right] - \log \frac{\pi_0}{\pi_1}$$

$$\sigma(z) = \sigma(\vec{\omega} \cdot \vec{x} + b)$$

$$\vec{\omega} = \Sigma^{-1}(\vec{\mu_1} - \vec{\mu_0})^T \quad b = -\log \frac{\pi_0}{\pi_1} + \frac{1}{2}\Sigma^{-1}\left(\vec{\mu_0}^T \vec{\mu_0} - \vec{\mu_1}^T \vec{\mu_1}\right)$$

Softmax + cross entropy ← classifier
↳ compares 2 PDFs

c) $\frac{P(y=1|\vec{x})}{P(y=0|\vec{x})} = \frac{\sigma(w\vec{x}+b)}{1-\sigma(w\vec{x}+b)} = \frac{\sigma(w\vec{x}+b)}{\sigma(-w\vec{x}-b)}$

$$= \frac{\frac{1}{1+e^{-(\vec{w}.\vec{x}+b)}}}{\frac{1}{1+e^{+(\vec{w}.\vec{x}+b)}}} = \frac{1+e^{\vec{w}.\vec{x}+b}}{1+e^{-(\vec{w}.\vec{x}+b)}} = K.$$

$z = \vec{w}.\hat{x}+b$ $= \frac{1+e^{z}}{1+e^{-z}} = \frac{e^{z}.e^{-z}+e^{z}}{1+e^{-z}}$

$$= \frac{e^{z}(\cancel{1+e^{-z}})}{\cancel{1+e^{-z}}}$$

$e^{z} = K.$ $\leftarrow$ $= e^{z} = K.$

$\log K = z = \vec{w}.\hat{x}+b$ is the hyperplane

Boundary: $\vec{w}.\vec{x}+b = \log 1 = 0. \rightarrow$ OK!!

Question 3:

a) $\mathcal{L}(\hat{v}) = \sum_{t=1}^{T} \log P(y_t|\vec{x}_t)$

$$= \sum_{t=1}^{T} y_t \log P(y_t=1/x_t) + (1-y_t)\log P(y_t=0/\vec{x}_t).$$

$$= \sum_{t=1}^{T} y_t \log(1-e^{-\vec{v}.\vec{x}}) + (1-y_t)\log(1-(1-e^{-\vec{v}.\vec{x}}))$$

$$= \sum_{t=1}^{T} y_t \log\left(1 - e^{-\vec{v}.\vec{x}}\right) + (1-y_t). \log e^{-\vec{v}.\vec{x}}$$

$$= \sum_{t=1}^{T} y_t \log\left(1 - e^{-\vec{v}.\vec{x}}\right) - (1-y_t)\,\vec{v}.\vec{x}.$$

b) $$\frac{\partial \mathcal{L}}{\partial \vec{v}} = \sum_{t=1}^{T} \frac{y_t\left(-e^{-\vec{v}.\vec{x}}\right)(-\vec{x})}{1 - e^{-\vec{v}.\vec{x}}} - (1-y_t).x_t$$

$$= \sum_{t=1}^{T} \vec{x}_t \left(\frac{y_t e^{-\vec{v}.\vec{x}}}{p_t} - \frac{p_t}{p_t} + \frac{y_t p_t}{p_t} \right)$$

$$= \sum_{t=1}^{T} \vec{x}_t \left(y_t \left(\frac{e^{-\vec{v}.\vec{x}} + (1-e^{-\vec{v}.\vec{x}})}{p_t} \right) - \frac{p_t}{p_t} \right)$$

$$= \sum_{t=1}^{T} \vec{x}_t \left[\frac{y_t - p_t}{p_t} \right] \quad \text{expression !!}$$

c) $$P(y=1|\vec{x}) = 1 - e^{-\vec{v}.\vec{x}}$$

$$= 1 - e^{-(-\vec{x}\log(1-\vec{p}))}$$

$$= 1 - e^{\log(1-\vec{p})\vec{x}}$$

$$= 1 - e^{\sum_{i=1}^{k} \log(1-p_i)x_i}$$

$$= 1 - e^{\log \prod_{i=1}^{k} (1-p_i)^{x_i}}$$

$$= 1 - \prod_{i=1}^{k} (1-p_i)^{x_i}$$

d) $\frac{\partial L}{\partial p_i} = \frac{\partial L}{\partial v_i} \cdot \frac{\partial v_i}{\partial p_i}$

$\frac{\partial v_i}{\partial p_i} = \frac{\partial}{\partial p_i}\left(-\log(1-p_i)\right) = \frac{1}{1-p_i}$ $\Rightarrow \boxed{\frac{\partial L}{\partial p_i} = \frac{1}{1-p_i} \cdot \frac{\partial \mathcal{L}}{\partial v_i}}$

e) $p_i \leftarrow p_i + \eta_i \frac{\partial \mathcal{L}}{\partial p_i}$

$$p_i + \eta_i \frac{\partial \mathcal{L}}{\partial p_i} = p_i + \frac{p_i \cancel{(1-p_i)}}{T_i} \cdot \frac{1}{\cancel{1-p_i}} \frac{\partial \mathcal{L}}{\partial v_i}$$

$$= p_i + \frac{p_i \, \partial \mathcal{L}}{T_i \, \partial v_i}$$

$$= p_i + \frac{p_i}{T_i} \sum_{t=1}^{T} x_{it} \left[\frac{y_t - p_t}{p_t}\right]$$

$$= p_i + \frac{p_i}{T_i} \sum_{t=1}^{T} \frac{x_{it} \cdot y_t - x_{it} p_t}{p_t}$$

$$= p_i + \frac{p_i}{T_i} \sum_{t=1}^{T} \frac{x_{it} \cdot y_t}{p_t} - \frac{p_i}{T_i} \sum_{t=1}^{T} x_{it}$$

$$= \cancel{p_i} + \frac{p_i}{T_i} \sum_{t=1}^{T} \frac{x_{it} \cdot y_t}{p_t} - \cancel{\frac{p_i}{T_i} \cdot T_i}$$

$$= \frac{p_i}{T_i} \sum_{t=1}^{T} \frac{x_{it} \cdot y_t}{p_t}.$$

Question 4:

a) $P(y=1, y'=1 \mid \vec{x}, \vec{x}', s=1)$

Bayes

$$= \frac{P(s=1 \mid y=1, y'=1, \vec{x}, \vec{x}') \cdot P(y=1, y'=1 \mid \vec{x}, \vec{x}')}{P(s=1 \mid \vec{x}, \vec{x}')}$$

CI & Product Rule

$$= \frac{P(s=1 \mid y=1, y'=1) \cdot P(y=1 \mid \vec{x}, \vec{x}') \cdot P(y'=1 \mid \vec{x}, \vec{x}', y=1)}{P(s=1 \mid \vec{x}, \vec{x}')}$$

CI

$$= \frac{P(s=1 \mid y=1, y'=1) \cdot P(y=1 \mid \vec{x}) \cdot P(y'=1 \mid \vec{x}')}{P(s=1 \mid \vec{x}, \vec{x}')}$$

PR

$$= \frac{1 \cdot \sigma(\vec{w} \cdot \vec{x}) \cdot \sigma(\vec{w} \cdot \vec{x}')}{P(s=1 \mid \vec{x}, \vec{x}')} \overset{\text{marg}}{=} \frac{1 \cdot \sigma(\vec{w} \cdot \vec{x}) \cdot \sigma(\vec{w} \cdot \vec{x}')}{\sum_{y,y'} P(s=1, y, y' \mid \vec{x}, \vec{x}')}$$

marg

$$= \frac{1 \cdot \sigma(\vec{w} \cdot \vec{x}) \cdot \sigma(\vec{w} \cdot \vec{x}')}{\sum_{y,y'} P(s=1, y, y' \mid \vec{x}, \vec{x}')}$$

$$= \frac{1 \cdot \sigma(\vec{w} \cdot \vec{x}) \cdot \sigma(\vec{w} \cdot \vec{x}')}{\sum_{y,y'} P(s=1 \mid y, y') \cdot P(y \mid \vec{x}) \cdot P(y' \mid \vec{x}')}$$

$$= \frac{1 . \sigma(\vec{\omega}.\vec{x}) . \sigma(\vec{\omega}.\vec{x'})}{\sum_{y,y'} I(y,y') . P(y|\vec{x}) . P(y'|\vec{x'})}$$

$$= \frac{\sigma(\vec{\omega}.\vec{x}) . \sigma(\vec{\omega}.\vec{x'})}{P(y=0|\vec{x}) . P(y'=0|\vec{x'}) + P(y=1|\vec{x}) . P(y'=1|\vec{x'})}$$

$$= \frac{\sigma(\vec{\omega}.\vec{x}) . \sigma(\vec{\omega}.\vec{x'})}{(1-\sigma(\vec{\omega}.\vec{x}))(1-\sigma(\vec{\omega}.\vec{x'})) + \sigma(\vec{\omega}.\vec{x}) . \sigma(\vec{\omega}.\vec{x'})}$$

$$= \frac{\sigma(\vec{\omega}.\vec{x}) . \sigma(\vec{\omega}.\vec{x'})}{\sigma(-\vec{\omega}.\vec{x}) . \sigma(-\vec{\omega}.\vec{x'}) + \sigma(\vec{\omega}.\vec{x}) . \sigma(\vec{\omega}.\vec{x'})}$$

b) $P(y=1, y'=0 \,|\, \vec{x}, \vec{x'}, s=0)$

$$= \frac{P(s=0 \,|\, y=1, y'=0, \vec{x}, \vec{x}) . P(y=1, y'=0 \,|\, \vec{x}, \vec{x'})}{P(s=0 \,|\, \vec{x}, \vec{x'})}$$

CI
& PR.

$$= \frac{P(s=0 \,|\, y=1, y'=0) . P(y=1 \,|\, \vec{x}, \vec{x}) . P(y'=0 \,|\, \vec{x}, \vec{x'}, y=1)}{P(s=0 \,|\, \vec{x}, \vec{x'})}$$

$$= \frac{P(s=0 \,|\, y=1, y'=0) . P(y=1 \,|\, \vec{x}) . P(y'=0 \,|\, \vec{x'})}{P(s=0 \,|\, \vec{x}, \vec{x'})}$$

$$= \frac{1 . \sigma(\vec{\omega}.\vec{x}) . \sigma(-\vec{\omega}.\vec{x'})}{P(s=1 \,|\, \vec{x}, \vec{x'})}$$

$$= \frac{\sigma(\vec{w}\cdot\vec{x})\cdot\sigma(-\vec{w}\cdot\vec{x'})}{P(s=1|\vec{x},\vec{x'})}$$

$$= \frac{\sigma(\vec{w}\cdot\vec{x})\cdot\sigma(-\vec{w}\cdot\vec{x'})}{\sum_{y,y'} P(s=0,y,y'|\vec{x},\vec{x'})}$$

$$= \frac{\sigma(\vec{w}\cdot\vec{x})\cdot\sigma(-\vec{w}\cdot\vec{x'})}{\sum_{y,y'} P(s=0|y,y')\cdot P(y|\vec{x})\cdot P(y'|\vec{x'})}$$

$$= \frac{\sigma(\vec{w}\cdot\vec{x})\cdot\sigma(-\vec{w}\cdot\vec{x'})}{P(y=0|\vec{x})\cdot P(y'=1|\vec{x'}) + P(y=1|\vec{x})\cdot P(y'=0|\vec{x'})}$$

$$= \frac{\sigma(\vec{w}\cdot\vec{x})\cdot\sigma(-\vec{w}\cdot\vec{x'})}{\sigma(-\vec{w}\cdot\vec{x})\cdot\sigma(\vec{w}\cdot\vec{x'}) + \sigma(\vec{w}\cdot\vec{x})\cdot\sigma(-\vec{w}\cdot\vec{x'})}$$

c)

i - b	$P(y=1,y'=1|\vec{x},\vec{x'},s=1)$
ii - b	$P(y=1,y'=1|\vec{x},\vec{x'},s=1)$
iii - a	$P(y=1,y'=0|\vec{x},\vec{x'},s=0)$
iv - c	$1 - P(y=1,y'=0|\vec{x},\vec{x'},s=0)$

Scanned with CamScanner

d) $\mathcal{L}(\vec{w}) = \sum_t \log P(s_t \mid \vec{x_t}, \vec{x_t'})$

$= \sum_t s_t \log P(s=1 \mid \vec{x_t}, \vec{x_t'}) + (1-s_t) . \log P(s=0 / \vec{x_t}, \vec{x_t'})$

from a/b $= \sum_t s_t \log\left(\sigma(-\vec{w}.\vec{x}) . \sigma(\vec{w}.\vec{x'}) + \sigma(\vec{w}.\vec{x}) . \sigma(\vec{w}, \vec{x'})\right)$

$+ (1-s_t) \log\left(\sigma(-\vec{w}.\vec{x'}) . \sigma(\vec{w}.\vec{x'}) + \sigma(\vec{w}.\vec{x'}) . \sigma(-\vec{w}.\vec{x'})\right)$

e) $\vec{w} \leftarrow \vec{w} + \eta \left\{ \sum_t \left[(\vec{y_t} - \sigma(\vec{w}.\vec{x_t}))\vec{x_t} + (\vec{y_t'} - \sigma(\vec{w}.\vec{x_t'}))\vec{x_t'} \right] \right.$

Final answer

Explaination:

$L(\vec{w}) = \sum_t \left(\vec{y_t} \log \sigma(\vec{w}.\vec{x_t}) + (1-\vec{y_t}) \log \sigma(-\vec{w}.\vec{x_t}) + \vec{y_t'} . \log \sigma(\vec{w}.\vec{x_t'}) + (1-\vec{y_t'}) \log \sigma(-\vec{w}.\vec{x_t'}) \right)$

$\frac{\partial L}{\partial w} = \sum_t \left(\frac{\vec{y_t} . \sigma(\vec{w}.\vec{x_t}) . \sigma(-\vec{w}.\vec{x_t})\vec{x_t}}{\sigma(\vec{w}.\vec{x_t})} - \frac{(1-\vec{y_t}) . \sigma(-\vec{w}.\vec{x_t})\sigma(\vec{w}.\vec{x_t}) . \vec{x_t}}{\sigma(-\vec{w}.\vec{x_t})} + \frac{\vec{y_t'} \sigma(\vec{w}.\vec{x_t'}) . \sigma(-\vec{w}.\vec{x_t'}) . \vec{x_t'}}{\sigma(\vec{w}.\vec{x_t'})} - \left[\frac{(1-\vec{y_t'}) . \sigma(-\vec{w}.\vec{x_t'}) . \sigma(\vec{w}.\vec{x_t'}) . \vec{x_t'}}{\sigma(-\vec{w}.\vec{x_t'})} \right. \right.$

Scanned with CamScanner

$$\cancel{\sigma'(z) = \sigma(z).\sigma(-z)}$$

$$\sum_t \Big[\vec{y_t} . \sigma(-\vec{\omega}.\vec{x_t}).\vec{x_t} - (1-\vec{y_t}).\sigma(\vec{\omega}.\vec{x_t}).\vec{x_t}$$
$$+ \vec{y_{t'}} . \sigma(-\vec{\omega}.\vec{x_{t'}}).\vec{x_{t'}} - (1-\vec{y_{t'}}) \times \sigma(\vec{\omega}.\vec{x_{t'}}).\vec{x_{t'}} \Big]$$

$$= \underbrace{\sum_t \Big[\vec{y_t} - \sigma(\vec{\omega}.\vec{x_t}).\vec{x_t} \Big] \vec{x_t} + \Big[\vec{y_{t'}} - \sigma(\vec{\omega}.\vec{x_{t'}}) \Big] \vec{x_{t'}}}$$

$$\Rightarrow \vec{\omega} \leftarrow \vec{\omega} + \eta \Big[\quad \Downarrow \quad \Big]$$

proub.

HW8_250A_A59020215

November 29, 2022

[ ]:
```
import numpy as np
from collections import defaultdict

!unzip /content/hw8files.zip -d /content
```

```
Archive:  /content/hw8files.zip
replace /content/__MACOSX/._hw8files? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
  inflating: /content/__MACOSX/._hw8files
  inflating: /content/hw8files/hw8_probR_init.txt
  inflating: /content/__MACOSX/hw8files/._hw8_probR_init.txt
  inflating: /content/hw8files/hw8_movies.txt
  inflating: /content/__MACOSX/hw8files/._hw8_movies.txt
  inflating: /content/hw8files/hw8_ratings.txt
  inflating: /content/__MACOSX/hw8files/._hw8_ratings.txt
  inflating: /content/hw8files/hw8_probZ_init.txt
  inflating: /content/__MACOSX/hw8files/._hw8_probZ_init.txt
  inflating: /content/hw8files/hw8_ids.txt
  inflating: /content/__MACOSX/hw8files/._hw8_ids.txt
```

[ ]:
```
# /content/hw8files/hw8_ids.txt
users = np.loadtxt("hw8_ids.txt",dtype=str)
items = np.loadtxt("hw8_movies.txt",dtype=str)
Rt = np.loadtxt("hw8_ratings.txt",dtype=str)
```

[ ]:
```
print(Rt[0,2] == "?")
```

```
True
```

[ ]:
```
ratings = np.zeros((len(Rt),len(Rt[0])))
for i in range(len(Rt)):
  for j in range(len(Rt[0])):
    if Rt[i][j] != '?':
      ratings[i][j] = int(Rt[i,j])
    else:
      ratings[i,j] = -1
```

[ ]:
```
Interactions = []

for i in range(len(ratings)):
```

```
  for j in range(len(ratings[0])):
    Interactions.append((users[i],items[j],ratings[i][j]))
```

[ ]:
```
Interactions[:10]
```

[ ]:
```
[('\ufeffA12614795', 'Inception', 1.0),
 ('\ufeffA12614795', 'The_Social_Network', 1.0),
 ('\ufeffA12614795', 'Black_Swan', -1.0),
 ('\ufeffA12614795', 'Shutter_Island', 1.0),
 ('\ufeffA12614795', 'The_Last_Airbender', 0.0),
 ('\ufeffA12614795', 'Harry_Potter_and_the_Deathly_Hallows:_Part_1', -1.0),
 ('\ufeffA12614795', 'Iron_Man_2', 1.0),
 ('\ufeffA12614795', 'Toy_Story_3', 1.0),
 ('\ufeffA12614795', 'Fast_Five', 1.0),
 ('\ufeffA12614795', 'Thor', 1.0)]
```

[ ]:
```
ratingsPerUser = defaultdict(list)
ratingsPerItem = defaultdict(list)
usersPerItem = defaultdict(set)
itemsPerUser = defaultdict(set)
ratingDict = {}

for interaction in Interactions:
    ratingsPerUser[interaction[0]].append((interaction[1],interaction[2]))
    ratingsPerItem[interaction[1]].append((interaction[0],interaction[2]))
    itemsPerUser[interaction[0]].add(interaction[1])
    usersPerItem[interaction[1]].add(interaction[0])
    ratingDict[(interaction[0],interaction[1])] = interaction[2]
```

[ ]:
```
Dict = {}
for item in ratingsPerItem:
  liked = 0
  seen = 0
  for user in ratingsPerUser:
    rating = ratingDict[(user,item)]
    if rating == 1:
      liked+=1
    if rating != -1:
      seen+=1
  if seen==0:
    Dict[item] = -1
  else:
    Dict[item] = liked/seen
```

[ ]:
```
popular_Dict = dict(sorted(Dict.items(),key=lambda item:item[1]))
```

#Q1-A

[ ]: popular_Dict

```
[ ]: {'Chappaquidick': 0.34285714285714286,
     'The_Last_Airbender': 0.460431654676259,
     'I_Feel_Pretty': 0.4878048780487805,
     'Fifty_Shades_of_Grey': 0.4975124378109453,
     'Fast_&_Furious:_Hobbs_&_Shaw': 0.5181347150259067,
     'Hustlers': 0.5185185185185185,
     'Magic_Mike': 0.5333333333333333,
     'Bridemaids': 0.5384615384615384,
     'World_War_Z': 0.5774647887323944,
     'The_Shape_of_Water': 0.5795454545454546,
     'Good_Boys': 0.6,
     'Prometheus': 0.6017699115044248,
     'Pokemon_Detective_Pikachu': 0.6020942408376964,
     'American_Hustle': 0.6052631578947368,
     'Terminator:_Dark_Fate': 0.6052631578947368,
     'The_Farewell': 0.6111111111111112,
     'Man_of_Steel': 0.625,
     'Fast_Five': 0.6256684491978609,
     'The_Hateful_Eight': 0.6282051282051282,
     'Star_Wars:_The_Force_Awakens': 0.6495726495726496,
     'The_Help': 0.6515151515151515,
     'Rocketman': 0.6515151515151515,
     'Drive': 0.6623376623376623,
     'The_Girls_with_the_Dragon_Tattoo': 0.6777777777777778,
     'Thor': 0.6804123711340206,
     'Avengers:_Age_of_Ultron': 0.6928104575163399,
     'Phantom_Thread': 0.6976744186046512,
     'Us': 0.6976744186046512,
     'The_Revenant': 0.7007874015748031,
     'X-Men:_First_Class': 0.701195219123506,
     'Pitch_Perfect': 0.7086614173228346,
     'Dunkirk': 0.7135416666666666,
     'Ready_Player_One': 0.7142857142857143,
     'Room': 0.7162162162162162,
     'Jurassic_World': 0.7171717171717171,
     'Mad_Max:_Fury_Road': 0.7243589743589743,
     'Once_Upon_a_Time_in_Hollywood': 0.725925925925926,
     'Manchester_by_the_Sea': 0.7272727272727273,
     'The_Perks_of_Being_a_Wallflower': 0.7346938775510204,
     'Spiderman:_Far_From_Home': 0.7450331125827815,
     'Her': 0.75,
     'Captain_America:_The_First_Avenger': 0.7523510971786834,
     'Frozen': 0.7552447552447552,
     'Hidden_Figures': 0.7590361445783133,
     'The_Hunger_Games': 0.7676767676767676,
```

```
 'Iron_Man_2': 0.770392749244713,
 'Les_Miserables': 0.7737226277372263,
 'Toy_Story_3': 0.7777777777777778,
 'Three_Billboards_Outside_Ebbing': 0.7804878048780488,
 'Darkest_Hour': 0.78125,
 'Ex_Machina': 0.7863247863247863,
 'Gone_Girl': 0.7938931297709924,
 'Black_Swan': 0.7981651376146789,
 '12_Years_a_Slave': 0.811965811965812,
 'Avengers:_Endgame': 0.8136645962732919,
 'The_Avengers': 0.8147058823529412,
 'Midnight_in_Paris': 0.8170731707317073,
 'The_Great_Gatsby': 0.8178137651821862,
 'La_La_Land': 0.8235294117647058,
 'Avengers:_Infinity_War': 0.826625386996904,
 'The_Theory_of_Everything': 0.8294117647058824,
 'Now_You_See_Me': 0.836,
 '21_Jump_Street': 0.837037037037037,
 'Django_Unchained': 0.84472049689441,
 'The_Martian': 0.861244019138756,
 'Harry_Potter_and_the_Deathly_Hallows:_Part_1': 0.8628048780487805,
 'Joker': 0.8647686832740213,
 'Wolf_of_Wall_Street': 0.8669064748201439,
 'The_Lion_King': 0.8717948717948718,
 'Harry_Potter_and_the_Deathly_Hallows:_Part_2': 0.871875,
 'Parasite': 0.8878923766816144,
 'The_Social_Network': 0.8960396039603961,
 'The_Dark_Knight_Rises': 0.8982456140350877,
 'Shutter_Island': 0.9333333333333333,
 'Interstellar': 0.946843853820598,
 'Inception': 0.9605263157894737}
```

I would say that they somewhat align, with a few exceptions here and there. But given the big size of the class, taking all these samples, would result in a reflective ranking which explains why it aligns with my personal preferences.

#Q1-E

[ ]:
```
Pz = np.loadtxt("hw8_probZ_init.txt")
PRgZ = np.loadtxt("hw8_probR_init.txt")
```

[ ]:
```
users[:10]
```

[ ]:
```
array(['\ufeffA12614795', 'A59036954', 'A13991559', 'A16043858',
       'A59017523', 'A59019971', 'A59012875', 'A16343698', 'A59019732',
       'A59020506'], dtype='<U15')
```

[ ]:
```
print(Pz.shape)
print(PRgZ.shape)
k = 4
N = 256
posterior = np.zeros((k,len(users)))
```

```
(4,)
(76, 4)
```

[ ]:
```
def loglikeli(user):
  sumi = 0
  liked = np.where(ratings[user,:] == 1)
  disliked = np.where(ratings[user,:] == 0)
  for i in range(k):
    sumi+= Pz[i]*(np.prod(PRgZ[liked,i]))*(np.prod(1-PRgZ[disliked,i]))
  return np.log(sumi)
```

[ ]:
```
def E_step(i,user):
  liked = np.where(ratings[user,:] == 1)
  disliked = np.where(ratings[user,:] == 0)
  deno = 0
  for ind in range(k):
    deno+=Pz[ind]*np.prod(PRgZ[liked,ind])*(np.prod(1-PRgZ[disliked,ind]))
  return Pz[i]*np.prod(PRgZ[liked,i])*(np.prod(1-PRgZ[disliked,i]))/deno
```

[ ]:
```
def M_step(i,item):
  seen = np.where(ratings[:,item] == 1)
  notSeen = np.where(ratings[:,item] == -1)
  sumi = (np.sum(posterior[i,seen])+PRgZ[item,i]*np.sum(posterior[i,notSeen]))/
 ↪np.sum(posterior[i,:])
  return sumi
```

[ ]:
```
print("Iteration No ", "\t" , "LogLikelihood")
printed = [2**i for i in range(9) ]
printed.insert(0,0)

for i in range(N+1):
  logli = 0
  TmpZ = np.zeros(k)
  TmpRZ = np.zeros((len(items),k))
  for user in range(len(users)):
    logli += loglikeli(user)
    for j in range(k):
      posterior[j,user] = e_step(j,user)
  for l in range(k):
    TmpZ[l] = np.sum(posterior[l,:])/len(users)
    for item in range(len(items)):
      TmpRZ[item,l] = m_step(l,item)
```

```
    Pz  = TmpZ
    PRgZ = TmpRZ
    if i in printed:
      print(i,"\t" ,"\t" , round(logli/len(users),3))
```

```
Iteration No      LogLikelihood
0                 -28.627
1                 -19.35
2                 -17.91
4                 -17.081
8                 -16.63
16                -16.288
32                -15.802
64                -15.75
128               -15.736
256               -15.729
```

They are getting bigger, by that I mean is less negative and it is a trend we gave been seeing even in hw6.

#Q1-F

[ ]:
```
user = "A59020215"
```

[ ]:
```
ID = np.where(users == user)[0][0]
```

[ ]:
```
recs = {}
for elem in ratingsPerUser[user]:
  if elem[1] == -1:
    sumi = 0
    movieID = np.where(items==elem[0])[0][0]
    for i in range(k):
      sumi += e_step(i,ID)*m_step(i,movieID)
    recs[elem[0]] = sumi
```

[ ]:
```
My_recommendations = dict(sorted(recs.items(),key=lambda item:
 ↪item[1],reverse=True))
```

[ ]:
```
My_recommendations
```

[ ]:
```
{'Inception': 0.9997710233967291,
 'Interstellar': 0.8324232158141902,
 'The_Social_Network': 0.7058685788466552,
 'Us': 0.6653382792558556,
 'Her': 0.4938857369788067,
 'The_Dark_Knight_Rises': 0.4584038119980221,
 'Mad_Max:_Fury_Road': 0.43584616736490994,
 '21_Jump_Street': 0.43010396779340926,
 'Midnight_in_Paris': 0.375299480858295,
```

```
 'The_Girls_with_the_Dragon_Tattoo': 0.33994221885756915,
 'Fast_&_Furious:_Hobbs_&_Shaw': 0.3396832424189142,
 'The_Shape_of_Water': 0.33842030617609187,
 'Rocketman': 0.3344074634184864,
 'Three_Billboards_Outside_Ebbing': 0.31842873200562,
 'Pitch_Perfect': 0.3069615781755432,
 'Terminator:_Dark_Fate': 0.28833927088407957,
 'The_Perks_of_Being_a_Wallflower': 0.2865016906339786,
 'Good_Boys': 0.2514692250568793,
 'Phantom_Thread': 0.2492949290350923,
 'Dunkirk': 0.23834100527481833,
 'X-Men:_First_Class': 0.17072468452282433,
 'Magic_Mike': 0.15145570233947217,
 'Pokemon_Detective_Pikachu': 0.11568210358761286,
 'Parasite': 0.09680703389120608,
 'Drive': 0.0017148438876369411,
 'Bridemaids': 0.0013885982644431306}
```

Yes, the recommendations are good, especially with the 'similarity' score which makes them much more real. I will have to take look at inception because it was the people's favorite and I hadn't watched it yet. It was a fun problem to code and it was really insightful.

[ ]:

```

```