HW8 CSE250 SectionA

Name: Guanghao Chen

PID: A53276390

Email: guc001 at eng.ucsd.edu

8.1 EM algorithm for binary matrix completion

(a)

This order list of movies not similar to my preferences.

```
The_Last_Airbender
Fifty_Shades_of_Grey
I_Feel_Pretty
Chappaquidick
Man_of_Steel
Prometheus
The_Shape_of_Water
Phantom_Thread
Magic_Mike
World_War_Z
Bridemaids
American_Hustle
Drive
The_Hunger_Games
Thor
Pitch_Perfect
Fast_Five
Avengers:_Age_of_Ultron
Jurassic_World
The_Hateful_Eight
The_Revenant
Dunkirk
Star_Wars:_The_Force_Awakens
Mad_Max:_Fury_Road
Captain_America:_The_First_Avenger
The_Perks_of_Being_a_Wallflower
Iron_Man_2
La_La_Land
Manchester_by_the_Sea
The_Help
```

```
Midnight_in_Paris
The_Girls_with_the_Dragon_Tattoo
21_Jump_Street
Frozen
Now_You_See_Me
X-Men:_First_Class
Ex_Machina
Harry_Potter_and_the_Deathly_Hallows:_Part_1
Toy_Story_3
Her
The_Great_Gatsby
The_Avengers
The_Theory_of_Everything
Room
Gone_Girl
Three_Billboards_Outside_Ebbing
Les_Miserables
Harry_Potter_and_the_Deathly_Hallows:_Part_2
The_Martian
Avengers:_Infinity_War
Darkest_Hour
Hidden_Figures
12_Years_a_Slave
Ready_Player_One
Black_Swan
Django_Unchained
Wolf_of_Wall_Street
Shutter_Island
Interstellar
The_Dark_Knight_Rises
The_Social_Network
Inception
```

(b)

$$
\begin{aligned}
&P(\{R_j = r_j^t\}_{j\in\Omega_t}) \\
&= \sum_{i=1}^{k} P(Z=i, \{R_j = r_j^t\}_{j\in\Omega_t}) \\
&= \sum_{i=1}^{k} P(Z=i) P(\{R_j = r_j^t\}_{j\in\Omega_t} | Z=i) \\
&= \sum_{i=1}^{k} P(Z=i) P(\{R_j = r_j^t\}_{j\in\Omega_t} | Z=i) \\
&= \sum_{i=1}^{k} P(Z=i) \prod_{j\in\Omega_t} P(R_j = r_j^t | Z=i)
\end{aligned}
$$

(c)

$$
\begin{aligned}
&P(Z=i|\{R_j=r_j^t\}_{j\in\Omega_t}) \\
&= \frac{P(Z=i)P(\{R_j=r_j^t\}_{j\in\Omega_t}|Z=i)}{P(\{R_j=r_j^t\}_{j\in\Omega_t})} \\
&= \frac{P(Z=i)\prod_{j\in\Omega_t}P(R_j=r_j^t|Z=i)}{\sum_{i'=1}^{k}P(Z=i')\prod_{j\in\Omega_t}P(R_j=r_j^t|Z=i')}
\end{aligned}
$$

(d)

Note that variable Z is hidden and variables R_j are observed.

Therefore,

$$
\begin{aligned}
&P(Z=i) \\
&= \frac{1}{T}\sum_{t=1}^{T}P(Z=i)|\{R_j=r_j^t\}_{j\in\Omega_t}) \\
&= \frac{1}{T}\sum_{t=1}^{T}\rho_{it}
\end{aligned}
$$

$$
\begin{aligned}
&P(R_j=1|Z=i) \\
&= \frac{\sum_t P(R_j=1, Z=i|\{R_j=r_j^t\}_{j\in\Omega_t})}{\sum_t P(Z=i)|\{R_j=r_j^t\}_{j\in\Omega_t})} \\
&= \frac{\sum_t P(R_j=1, Z=i|\{R_j=r_j^t\}_{j\in\Omega_t})}{\sum_{t=1}^{T}\rho_{it}} \\
&= \frac{\sum_{\{t|j\in\Omega_t\}}\rho_{it}I(r_j^t,1)+\sum_{\{t|j\notin\Omega_t\}}\rho_{it}P(R_j=1|Z=i)}{\sum_{t=1}^{T}\rho_{it}}
\end{aligned}
$$

(e)

Iterations:0,log-likelihood:-26.678832965400435 Iterations:1,log-likelihood:-16.094668997711192 Iterations:2,log-likelihood:-14.287794027341253 Iterations:4,log-likelihood:-13.265082934492524 Iterations:8,log-likelihood:-12.847308711972167 Iterations:16,log-likelihood:-12.705998052491518 Iterations:32,log-likelihood:-12.640737126831329 Iterations:64,log-likelihood:-12.616074566973708 Iterations:128,log-likelihood:-12.591194247298994

(f)

This list seems to be more similar with my preference to some extent.

```
My preference for movie <Inception> might be 0.9954437773878834
My preference for movie <The_Dark_Knight_Rises> might be 0.9570888790882955
My preference for movie <Wolf_of_Wall_Street> might be 0.9337307304884398
```

```
My preference for movie <The_Social_Network> might be 0.9196822776484885
My preference for movie <Now_You_See_Me> might be 0.8899980713995929
My preference for movie <Django_Unchained> might be 0.889346787636565
My preference for movie <Shutter_Island> might be 0.8883747739633127
My preference for movie <Les_Miserables> might be 0.8848266148314107
My preference for movie <The_Theory_of_Everything> might be 0.8779581039049331
My preference for movie <Toy_Story_3> might be 0.8707688538645239
My preference for movie <Star_Wars:_The_Force_Awakens> might be
0.8629306178013846
My preference for movie <Manchester_by_the_Sea> might be 0.8585411568742193
My preference for movie <Ex_Machina> might be 0.8526107427742802
My preference for movie <Black_Swan> might be 0.8495287683510985
My preference for movie <Room> might be 0.8470592540489595
My preference for movie <Darkest_Hour> might be 0.8453031813495533
My preference for movie <The_Martian> might be 0.841389795891906
My preference for movie <Three_Billboards_Outside_Ebbing> might be
0.840447114564767
My preference for movie <Hidden_Figures> might be 0.8404069062907241
My preference for movie <12_Years_a_Slave> might be 0.8244188162869036
My preference for movie <Her> might be 0.8234966238031737
My preference for movie <Frozen> might be 0.8176480630193058
My preference for movie <Gone_Girl> might be 0.8068251195379383
My preference for movie <Mad_Max:_Fury_Road> might be 0.8038798649804457
My preference for movie <Jurassic_World> might be 0.7977834486684239
My preference for movie <Pitch_Perfect> might be 0.7865310175162806
My preference for movie <The_Perks_of_Being_a_Wallflower> might be
0.7844037776881418
My preference for movie <21_Jump_Street> might be 0.7809131735151332
My preference for movie <The_Girls_with_the_Dragon_Tattoo> might be
0.7655288333858723
My preference for movie <The_Revenant> might be 0.740673074202572
My preference for movie <Dunkirk> might be 0.7389802406026169
My preference for movie <Midnight_in_Paris> might be 0.7386529301797399
My preference for movie <American_Hustle> might be 0.7028394900628858
My preference for movie <The_Hateful_Eight> might be 0.6940426064955122
My preference for movie <Drive> might be 0.6836336226914728
My preference for movie <Bridemaids> might be 0.6504224238302433
My preference for movie <Phantom_Thread> might be 0.6230701755469804
My preference for movie <Magic_Mike> might be 0.5700354347814061
My preference for movie <Prometheus> might be 0.5674060546063066
My preference for movie <I_Feel_Pretty> might be 0.5210211956749495
My preference for movie <Chappaquidick> might be 0.41956965096944837
My preference for movie <Fifty_Shades_of_Grey> might be 0.3817038499990061
My preference for movie <The_Last_Airbender> might be 0.30370767656716824
```

(g)

```python
with open('hw8_probZ_init.txt') as f:
    probZ = f.readlines()
for index,item in enumerate(probZ):
    probZ[index] = float(item.strip())
probZ = np.array(probZ)

with open('hw8_probRgivenZ_init.txt') as f:
    probRZ = f.readlines()
probRZM = []
for index,item in enumerate(probRZ[:-1]):
    temp = item.strip().split('   ')
    probRZM.append(list(map(lambda x:float(x),temp)))
probRZM = np.array(probRZM)

def post(zvalue,student,probZ,probRZM,ratingM):
    nume = probZ[zvalue]
    for j in range(62):
        if(ratingM[student,j]==1):
            nume *= probRZM[j,zvalue]
        elif(ratingM[student,j]==0):
            nume *= (1- probRZM[j,zvalue])
    demo = 0
    for k in range(4):
        temp = probZ[k]
        for j in range(62):
            if(ratingM[student,j]==1):
                temp *= probRZM[j,k]
            elif(ratingM[student,j]==0):
                temp *= (1- probRZM[j,k])
        demo += temp
    return nume/demo

def LFun(probZ,probRZM,ratingM):
    res = 0
    for t in range(269):
        mysum = 0
        for i in range(4):
            temp = probZ[i]
            for j in range(62):
                if(ratingM[t,j]==1):
                    temp *= probRZM[j,i]
                elif(ratingM[t,j]==0):
                    temp *= (1- probRZM[j,i])
            mysum += temp
        res += np.log(mysum)
    return res/269
```

```
check = [0,1,2,4,8,16,32,64,128]
for time in range(129):
    if time in check:
        print("Iterations:{},log-likelihood:
{}".format(time,LFun(probZ,probRZM,ratingM)))
    store = np.zeros((4,269))
    for i in range(4):
        for t in range(269):
            store[i,t] = post(i,t,probZ,probRZM,ratingM)

    for j in range(62):
        for i in range(4):
            newup = 0
            newdown = 0
            for t in range(269):
                newdown += store[i,t]
                if(ratingM[t,j] ==1):
                    newup += store[i,t]
                elif (ratingM[t,j]==-1):
                    newup += store[i,t]*probRZM[j,i]
            probRZM[j,i] = newup / newdown
    for i in range(4):
        newvalue = 0
        for t in range(269):
            newvalue += store[i,t]
        newvalue = newvalue / 269
        probZ[i] = newvalue
myGoer = []
for i in range(62):
    if(ratingM[68,i]==-1):
        res = 0
        for j in range(4):
            res += post(j,68,probZ,probRZM,ratingM) * probRZM[i,j]
        myGoer.append((i,res))
myGoer = sorted(myGoer,key=lambda x:x[1],reverse=True)
for i in myGoer:
    print("My preference for movie <{}> might be {}".format(movies[i[0]],i[1]))
```

8.2 Mixture model decision boundery

(a)

$$
\begin{aligned}
P(y=1|\vec{x}) &= \frac{P(\vec{x}|y=1)P(y=1)}{P(\vec{x})} \\
&= \frac{P(\vec{x}|y=1)P(y=1)}{\sum_{i=0}^{1} P(\vec{x}|y=i)P(y=i)} \\
&= \frac{\pi_1 (2\pi)^{-\frac{d}{2}} |\Sigma_1|^{-\frac{1}{2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_1)^T \Sigma_1^{-1}(\vec{x}-\vec{\mu}_1)}}{\sum_i \pi_i (2\pi)^{-\frac{d}{2}} |\Sigma_i|^{-\frac{1}{2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T \Sigma_i^{-1}(\vec{x}-\vec{\mu}_i)}} \\
&= \frac{\pi_1 |\Sigma_1|^{-\frac{1}{2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_1)^T \Sigma_1^{-1}(\vec{x}-\vec{\mu}_1)}}{\sum_i \pi_i |\Sigma_i|^{-\frac{1}{2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T \Sigma_i^{-1}(\vec{x}-\vec{\mu}_i)}}
\end{aligned}
$$

(b)

$$
\begin{aligned}
P(y=1|\vec{x}) &= \frac{\pi_1 |\sum|^{-\frac{1}{2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_1)^T \sum^{-1}(\vec{x}-\vec{\mu}_1)}}{\sum_{i=0}^{1} \pi_i |\sum|^{-\frac{1}{2}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^T \sum^{-1}(\vec{x}-\vec{\mu}_i)}} \\
&= \frac{1}{1+\frac{\pi_0}{\pi_1} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_0)^T \sum^{-1}(\vec{x}-\vec{\mu}_0)+\frac{1}{2}(\vec{x}-\vec{\mu}_1)^T \sum^{-1}(\vec{x}-\vec{\mu}_1)}} \\
&= \frac{1}{1+e^{log\frac{\pi_0}{\pi_1}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu}_0)^T \sum^{-1}(\vec{x}-\vec{\mu}_0)+\frac{1}{2}(\vec{x}-\vec{\mu}_1)^T \sum^{-1}(\vec{x}-\vec{\mu}_1)}} \\
&= \frac{1}{1+e^{log\frac{\pi_0}{\pi_1}-\frac{1}{2}(\vec{x}-\vec{\mu}_0)^T \sum^{-1}(\vec{x}-\vec{\mu}_0)+\frac{1}{2}(\vec{x}-\vec{\mu}_1)^T \sum^{-1}(\vec{x}-\vec{\mu}_1)}} \\
&= \frac{1}{1+e^{log\frac{\pi_1}{\pi_0}+[(u_1-u_0)^T \Sigma^{-1}\vec{x}+\frac{1}{2}u_0^T \Sigma^{-1} u_0-\frac{1}{2}u_1^T \Sigma^{-1} u_1]}} \\
\vec{w} &= (u_1-u_0)^T \Sigma^{-1} \\
b &= log\frac{\pi_1}{\pi_0}+\frac{1}{2}u_0^T \Sigma^{-1} u_0-\frac{1}{2}u_1^T \Sigma^{-1} u_1
\end{aligned}
$$

(c)

$$
\begin{aligned}
\frac{P(y=1|\vec{x})}{P(y=0|\vec{x})} &= k \\
P(y=1|\vec{x}) &= \frac{k}{k+1} \\
e^{-(\vec{w}\cdot\vec{x}+b)} &= \frac{1}{k} \\
\vec{w}\cdot\vec{x}+b &= \log(k)
\end{aligned}
$$

8.3 Gradient ascent vs EM

(a)

$$
\begin{aligned}
L(\vec{v}) &= \sum_{t=1}^{T} logP(y_t|\vec{x}_t) \\
&= \sum_{t=1}^{T} [y_t log(1-e^{-\vec{v}\cdot\vec{x}_t}) + (1-y_t) log(e^{-\vec{v}\cdot\vec{x}_t})]
\end{aligned}
$$

(b)

$$
\begin{aligned}
\frac{\partial L}{\partial \vec{v}} &= \sum [y_t \frac{1}{1-e^{-\vec{v}\cdot\vec{x}}} (-e^{-\vec{v}\cdot\vec{x}})(-\vec{x}) + (1-y_t)(-\vec{x})] \\
&= \sum (-\vec{x}) [\frac{y_t(-e^{-\vec{v}\cdot\vec{x}})}{1-e^{-\vec{v}\cdot\vec{x}}} + \frac{(1-y_t)(1-e^{-\vec{v}\cdot\vec{x}})}{1-e^{-\vec{v}\cdot\vec{x}}}] \\
&= \sum (-\vec{x}) [\frac{-y_t e^{-\vec{v}\cdot\vec{x}} + (1-e^{-\vec{v}\cdot\vec{x}} - y_t + y_t e^{-\vec{v}\cdot\vec{x}})}{1-e^{-\vec{v}\cdot\vec{x}}}] \\
&= \sum (-\vec{x}) [\frac{1-e^{-\vec{v}\cdot\vec{x}} - y_t}{1-e^{-\vec{v}\cdot\vec{x}}}] \\
&= \sum [\frac{y_t - \rho_t}{\rho_t}] \vec{x}
\end{aligned}
$$

(c)

$$
\begin{aligned}
P(y=1|\vec{x}) &= 1 - e^{-\vec{v}\cdot\vec{x}} \\
&= 1 - e^{-\sum v_i x_i} \\
&= 1 - e^{\sum -v_i x_i} \\
&= 1 - \prod e^{-v_i x_i} \\
&= 1 - \prod e^{log(1-p_i)x_i} \\
&= 1 - \prod e^{log(1-p_i)^{x_i}} \\
&= 1 - \prod (1-p_i)^{x_i}
\end{aligned}
$$

(d)

$$
\begin{aligned}
\frac{\partial v_i}{\partial p_i} &= \frac{1}{1-p_i} \\
L &= \sum_{t=1}^{T}[y_t log(1-e^{-\vec{v}\cdot\vec{x}_t}) + (1-y_t)log(e^{-\vec{v}\cdot\vec{x}_t})] \\
\frac{\partial L}{\partial p_i} &= \sum[y_t \frac{1}{1-e^{-\vec{v}\cdot\vec{x}}}(-e^{-\vec{v}\cdot\vec{x}})(-x_{it}) + (1-y_t)(-x_{it})]\frac{\partial v_i}{\partial p_i} \\
&= \sum(-x_{it})[\frac{y_t(-e^{-\vec{v}\cdot\vec{x}})}{1-e^{-\vec{v}\cdot\vec{x}}} + \frac{(1-y_t)(1-e^{-\vec{v}\cdot\vec{x}})}{1-e^{-\vec{v}\cdot\vec{x}}}]\frac{\partial v_i}{\partial p_i} \\
&= \sum(-x_{it})[\frac{-y_t e^{-\vec{v}\cdot\vec{x}} + (1-e^{-\vec{v}\cdot\vec{x}} - y_t + y_t e^{-\vec{v}\cdot\vec{x}})}{1-e^{-\vec{v}\cdot\vec{x}}}]\frac{\partial v_i}{\partial p_i} \\
&= \sum(-x_{it})[\frac{1-e^{-\vec{v}\cdot\vec{x}} - y_t}{1-e^{-\vec{v}\cdot\vec{x}}}]\frac{\partial v_i}{\partial p_i} \\
&= \sum[\frac{y_t-\rho_t}{\rho_t}]x_{it}\frac{1}{1-p_i} \\
&= \frac{\partial L}{\partial v_i}(\frac{1}{1-p_i})
\end{aligned}
$$

(e)

Substituting the result in part (b) and part(d) into the update rule of GA.

$$
\begin{aligned}
p_i + \eta(\frac{\partial L}{\partial v_i}) = p_i + \eta(\frac{\partial L}{\partial v_i}) \\
&= p_i + \eta(\frac{1}{1-p_i})(\sum_{t=1}^{T}[\frac{y_t-\rho_t}{\rho_t}]x_{it}) \\
&= p_i + \frac{p_i(1-p_i)}{T_i}(\frac{1}{1-p_i})(\sum_{t=1}^{T}[\frac{y_t-\rho_t}{\rho_t}]x_{it}) \\
&= p_i + \frac{p_i}{T_i}(\sum_{t=1}^{T}[\frac{y_t-\rho_t}{\rho_t}]x_{it}) \\
&= \frac{p_i T_i}{T_i} + \frac{p_i}{T_i}(\sum_{t=1}^{T}[\frac{y_t-\rho_t}{\rho_t}]x_{it}) \\
&= \frac{p_i\sum_{t=1}^{T}x_{it}}{T_i} + \frac{p_i\sum_{t=1}^{T}[\frac{y_t x_{it}}{\rho_t}] - p_i\sum_{t=1}^{T}x_{it}}{T_i} \\
&= \frac{p_i}{T_i}\sum_{t=1}^{T}[\frac{y_t x_{it}}{\rho_t}]
\end{aligned}
$$

8.4 Similarity learning with logistic regression

(a)

$$
\begin{aligned}
&P(y=1,y^{'}=1|\vec{x},\vec{x}^{'},s=1)\\
&=\frac{P(s=1|\vec{x},\vec{x}^{'},y=1,y^{'}=1)P(y=1,y^{'}=1|\vec{x},\vec{x}^{'})}{P(s=1|\vec{x},\vec{x}^{'})}\\
&=\frac{P(s=1|y=1,y^{'}=1)P(y=1|\vec{x},\vec{x}^{'})P(y^{'}=1|\vec{x},\vec{x}^{'})}{P(s=1,y=1,y^{'}=1|\vec{x},\vec{x}^{'})+P(s=1,y=0,y^{'}=0|\vec{x},\vec{x}^{'})}\\
&=\frac{P(s=1|y=1,y^{'}=1)P(y=1|\vec{x})P(y^{'}=1|\vec{x}^{'})}{P(s=1|y=1,y^{'}=1)P(y=1|\vec{x})P(y^{'}=1|\vec{x}^{'})+P(s=1|y=0,y^{'}=0)P(y=0|\vec{x})P(y^{'}=0|\vec{x}^{'})}\\
&=\frac{P(y=1|\vec{x},\vec{x}^{'})P(y^{'}=1|\vec{x},\vec{x}^{'})}{P(y=1|\vec{x})P(y^{'}=1|\vec{x}^{'})+P(y=0|\vec{x})P(y^{'}=0|\vec{x}^{'})}\\
&=\frac{\sigma(\vec{w}\cdot\vec{x})\sigma(\vec{w}\cdot\vec{x}^{'})}{\sigma(\vec{w}\cdot\vec{x})\sigma(\vec{w}\cdot\vec{x}^{'})+\sigma(-\vec{w}\cdot\vec{x})\sigma(-\vec{w}\cdot\vec{x}^{'})}
\end{aligned}
$$

(b)

$$
\begin{aligned}
&P(y=1,y^{'}=0|\vec{x},\vec{x}^{'},s=0)\\
&=\frac{P(s=0|\vec{x},\vec{x}^{'},y=1,y^{'}=0)P(y=1,y^{'}=0|\vec{x},\vec{x}^{'})}{P(s=0|\vec{x},\vec{x}^{'})}\\
&=\frac{P(s=0|y=1,y^{'}=1)P(y=1|\vec{x},\vec{x}^{'})P(y^{'}=1|\vec{x},\vec{x}^{'})}{P(s=0,y=1,y^{'}=1,\vec{x},\vec{x}^{'})+P(s=0,y=0,y^{'}=0,\vec{x},\vec{x}^{'})}\\
&=\frac{P(s=1|y=1,y^{'}=0)P(y=1|\vec{x},\vec{x}^{'})P(y^{'}=0|\vec{x},\vec{x}^{'})}{P(s=0|y=1,y^{'}=0)P(y=1|\vec{x},\vec{x}^{'})P(y^{'}=0|\vec{x},\vec{x}^{'})+P(s=0|y=1,y^{'}=0)P(y=1|\vec{x},\vec{x}^{'})P(y^{'}=0|\vec{x},\vec{x}^{'})}\\
&=\frac{P(y=1|\vec{x},\vec{x}^{'})P(y^{'}=0|\vec{x},\vec{x}^{'})}{P(y=1|\vec{x},\vec{x}^{'})P(y^{'}=0|\vec{x},\vec{x}^{'})+P(y=0|\vec{x},\vec{x}^{'})P(y^{'}=1|\vec{x},\vec{x}^{'})}\\
&=\frac{\sigma(\vec{w}\cdot\vec{x})\sigma(-\vec{w}\cdot\vec{x}^{'})}{\sigma(\vec{w}\cdot\vec{x})\sigma(-\vec{w}\cdot\vec{x}^{'})+\sigma(-\vec{w}\cdot\vec{x})\sigma(\vec{w}\cdot\vec{x}^{'})}
\end{aligned}
$$

(c)

(a)

(a)

(b)

(d)

(d)

$$
\begin{aligned}
L(\vec{w})&=\sum_t s_t logP(s=1|\vec{x}_t,\vec{x}_t^{'})+(1-s_t)P(s=0|\vec{x}_t,\vec{x}_t^{'})\\
&=\sum_t s_t log[\sigma(\vec{w}\cdot\vec{x})\sigma(\vec{w}\cdot\vec{x}^{'})+\sigma(-\vec{w}\cdot\vec{x})\sigma(-\vec{w}\cdot\vec{x}^{'})]\\
&+(1-s_t)log[\sigma(\vec{w}\cdot\vec{x})\sigma(-\vec{w}\cdot\vec{x}^{'})+\sigma(-\vec{w}\cdot\vec{x})\sigma(\vec{w}\cdot\vec{x}^{'})]
\end{aligned}
$$

(e)

The gradiant is

$$\frac{\partial L}{\partial \vec{w}} = \bar{y}_t \sigma(-\vec{w}\cdot\vec{x}_t)\vec{x}_t + (1-\bar{y}_t)\sigma(\vec{w}\cdot\vec{x}_t)(-\vec{x}_t) + \bar{y}_t^{'}\sigma(-\vec{w}\cdot\vec{x}_t^{'})\vec{x}_t^{'} + (1-\bar{y}_t^{'})\sigma(\vec{w}\cdot\vec{x}_t^{'})(-\vec{x}_t^{'})$$

8.5 Logistic regression across time

(a)

$$\begin{aligned}
\alpha_{j,\,t+1} &= P(Y_{t+1}=j|y_0,\overrightarrow{x_1},\ldots,\overrightarrow{x_{t+1}}) \\
&= \sum_i P(Y_t=i,Y_{t+1}=j|y_0,\overrightarrow{x_1},\ldots,\overrightarrow{x_{t+1}}) && (MA) \\
&= \sum_i P(Y_t=i|y_0,\overrightarrow{x_1},\ldots,\overrightarrow{x_{t+1}})P(Y_{t+1}=j|Y_t=i,y_0,\overrightarrow{x_1},\ldots,\overrightarrow{x_{t+1}}) && (PR) \\
&= \sum_i P(Y_t=i|y_0,\overrightarrow{x_1},\ldots,\overrightarrow{x_t})P(Y_{t+1}=j|Y_t=i,\overrightarrow{x_{t+1}}) && (CI) \\
&= \sum_i \alpha_{it} P(Y_{t+1}=j|Y_t=i,\overrightarrow{x_{t+1}}) \\
&= \sum_i \alpha_{it}[(1-Y_t)\sigma(\vec{\omega}_0\cdot\vec{x}_t)+Y_t\sigma(\vec{\omega}_1\cdot\vec{x}_t)]
\end{aligned}$$

(b)

$$\begin{aligned}
l^*_{jt+1} &= max_{y_1,\ldots,y_t}[logP(y_1,y_2,\ldots,y_{t+1}=j|y_0,\overrightarrow{x_1},\ldots,\overrightarrow{x_{t+1}})] \\
&= max_{y_1,\ldots,y_t}[logP(y_1,\ldots,y_t=i|y_0,\overrightarrow{x_1},\ldots,\overrightarrow{x_t})P(y_{t+1}=j|y_1,\ldots y_t=i,y_0,\overrightarrow{x_1},\ldots,\overrightarrow{x_{t+1}})] \\
&= max_{y_1,\ldots,y_t}[logP(y_1,\ldots,y_t=i|y_0,\overrightarrow{x_1},\ldots,\overrightarrow{x_t})+logP(y_{t+1}=j|y_t=i,\overrightarrow{x_{t+1}})] \\
&= max_{y_t}[max_{y_1,\ldots,y_{t-1}}logP(y_1,\ldots,y_t=i|y_0,\overrightarrow{x_1},\ldots,\overrightarrow{x_t})+logP(y_{t+1}=j|y_t=i,\overrightarrow{x_{t+1}})] \\
&= max_{y_t}[l^*_{it}+logP(y_{t+1}=j|y_t=i,\overrightarrow{x_{t+1}})] \\
&= max_{y_t}[l^*_{it}+log(j[(1-i)\sigma(\vec{\omega}_0\cdot\vec{x}_{t+1})+i\sigma(\vec{\omega}_1\cdot\vec{x}_{t+1})]+ \\
&\quad (1-j)[(1-i)(1-\sigma(\vec{\omega}_0\cdot\vec{x}_{t+1}))+i(1-\sigma(\vec{\omega}_1\cdot\vec{x}_{t+1}))])]
\end{aligned}$$

(c)

Therefore, the first blank is $\Phi_{t+1}(j) = argmax_{i\in\{0,1\}}[l^*_{it}+logP(y_{t+1}=j|y_t=i,\overrightarrow{x_{t+1}})]$

The second blank is $y^*_t = \Phi(y^*_{t+1})$