Course › Unit 4 Unsupervised Learning (2 weeks) › Project 4: Collaborative Filtering via Gaussian Mixtures › 7. Implementing EM for matrix completion

7. Implementing EM for matrix completion

We need to update our EM algorithm a bit to deal with the fact that the observations are no longer complete vectors. We use Bayes' rule to find an updated expression for the posterior probability $p(j|u) = P(y = j|x_{C_u}^{(u)})$:

$$p(j \mid u) = \frac{p(u|j)\cdot p(j)}{p(u)} = \frac{p(u|j)\cdot p(j)}{\sum_{j=1}^{K} p(u|j)\cdot p(j)} = \frac{\pi_j N(x_{C_u}^{(u)}; \mu_{C_u}^{(j)}, \sigma_j^2 I_{C_u\times C_u})}{\sum_{j=1}^{K} \pi_j N(x_{C_u}^{(u)}; \mu_{C_u}^{(j)}, \sigma_j^2 I_{C_u\times C_u})}$$

This is the soft assignment of cluster u to data point j.

To minimize numerical instability, you will be re-implementing the E-step in the log-domain, so you should calculate the values for the log of the posterior probability, $\ell(j,u) = \log(p(j|u))$ (though the actual output of your E-step should include the non-log posterior).

Let $f(u,i) = \log(\pi_i) + \log\left(N(x_{C_u}^{(u)}; \mu_{C_u}^{(i)}, \sigma_i^2 I_{C_u\times C_u})\right)$. Then, in terms of f, the log posterior is:

$$\begin{aligned}\ell\left(j|u\right) &= \log\left(p\left(j \mid u\right)\right) = \log\left(\frac{\pi_j N\left(x_{C_u}^{(u)};\mu_{C_u}^{(j)},\sigma_j^2 I_{C_u\times C_u}\right)}{\sum_{j=1}^{K}\pi_j N\left(x_{C_u}^{(u)};\mu_{C_u}^{(j)},\sigma_j^2 I_{C_u\times C_u}\right)}\right)\\
&= \log\left(\pi_j N\left(x_{C_u}^{(u)};\mu_{C_u}^{(j)},\sigma_j^2 I_{C_u\times C_u}\right)\right) - \log\left(\sum_{j=1}^{K}\pi_j N\left(x_{C_u}^{(u)};\mu_{C_u}^{(j)},\sigma_j^2 I_{C_u\times C_u}\right)\right)\\
&= \log\left(\pi_i\right) + \log\left(N\left(x_{C_u}^{(u)};\mu_{C_u}^{(i)},\sigma_i^2 I_{C_u\times C_u}\right)\right) - \log\left(\sum_{j=1}^{K}\exp\left(\log\left(\pi_j N\left(x_{C_u}^{(u)};\mu_{C_u}^{(j)},\sigma_j^2 I_{C_u\times C_u}\right)\right)\right)\right)\\
&= f\left(u,j\right) - \log\left(\sum_{j=1}^{K}\exp\left(f\left(u,j\right)\right)\right)\end{aligned}$$

Once we have evaluated $p\left(j|u\right)$ in the E-step, we can proceed to the M-step. We wish to find the parameters π, μ, and σ that maximize $\ell\left(X;\theta\right)$, the expected complete log-likelihood:

$$\ell\left(X;\theta\right) = \sum_{u=1}^{n}\left[\sum_{j=1}^{K} p\left(j|u\right)\log\left(\pi_j N\left(x_{C_u}^{(u)}|\mu_{C_u}^{(j)},\sigma_j^2 I_{|C_u|\times|C_u|}\right)\right)\right],$$

To maximize $\ell\left(X;\theta\right)$, we keep $p\left(j|u\right)$ (the soft-assignments) fixed, and maximize over the model parameters. Some of the parameters can be updated exactly as before with complete example vectors. For example,

$$\hat{\pi}_j = \frac{\sum_{u=1}^{n} p\left(j|u\right)}{n}$$

But we must be more careful in updating $\mu^{(j)}$ and σ_j^2 . This is because the parameters appear differently in the likelihood depending on how incomplete the observation is. Notice that some coordinates of $\mu^{(j)}$ do not impact observation $x_{C_u}^{(u)}$ at all. But we can proceed to separately update each coordinate of $\mu^{(j)}$.

The update equation for $\mu_i^{(j)}$ (the i^{th} coordinate of $\mu^{(j)}$) is derived as follows:

Decomposing the multivariate spherical Gaussians into univariate spherical Gaussians as before,

$$\begin{aligned} \ell\left(X;\theta\right) &= \sum_{u=1}^{n}\left[\sum_{j=1}^{K} p\left(j|u\right)\log\left(\pi_j\prod_{i=1}^{D}\left(N\left(x_i^{(u)}|\mu_i^{(j)},\sigma_{i,(j)}^2\right)\right)^{\delta(i,C_u)}\right)\right] \\ &= \sum_{u=1}^{n}\left[\sum_{j=1}^{K} p\left(j|u\right)\log\left(\pi_j\prod_{i=1}^{D}\left(\frac{1}{\sqrt{2\pi}\sigma_{i,(j)}}\exp\left(-\frac{1}{2\sigma_{i,(j)}^2}\left(x_i^{(u)}-\mu_i^{(j)}\right)^2\right)\right)^{\delta(i,C_u)}\right)\right] \end{aligned}$$

where $\delta\left(i,C_u\right)$ is an indicator function: 1 if $i \in C_u$ and zero otherwise.

Transforming the log of a product into a sum of logs and differentiating with respect to the lth movie coordinate for cluster j yields

$$\begin{aligned} \frac{\partial \ell(X;\theta)}{\partial \mu_l^{(j)}} &= \sum_{u=1}^{n} \left[p(j|u)\,\delta(l, C_u)\,(-\frac{1}{2\sigma_{l,(j)}^2}(-2x_l^{(u)} + 2\mu_l^{(j)})) \right] = 0 \\ 0 &= \sum_{u=1}^{n} p(j|u)\,\delta(l, C_u)\,(x_l^{(u)} - \mu_l^{(j)}) \\ \hat{\mu}_l^{(j)} &= \frac{\sum_{t=1}^{n} \delta(l, C_t)\,p(j \mid t)\,x_l^{(t)}}{\sum_{t=1}^{n} \delta(l, C_t)\,p(j \mid t)} \end{aligned}$$

We do **not** compute the mean update in the log domain; we use $p(j|u)$ instead of $\ell(j,u)$. When you set $\mu_i^{(j)}$ and σ_j^2 in the implementation, it will be easier, and not lead to numerical underflow issues, to use $p(j|u)$ instead of the logarithm $\ell(j,u)$.

Finally, the update equation for the variance is not too different from before:

$$\hat{\sigma}_j^2 = \frac{1}{\sum_{u=1}^{n} |C_u| p(j|u)} \sum_{u=1}^{n} p(j|u) \left\| x_{C_u}^{(u)} - \hat{\mu}_{C_u}^{(j)} \right\|^2$$

Implementation guidelines:

- You may find LogSumExp useful. But remember that your M-step should return the new $P = \hat{\pi}$, not the log of $\hat{\pi}$.
- The following will not affect the update equation above, but will affect your implementation: since we are dealing with incomplete data, we might have a case where most of the points in cluster j are missing the i-th coordinate. If we are not careful, the value of this coordinate in the mean will be determined by a small number of points, which leads to erratic

results. Instead, we should only update the mean when $\sum_{u=1}^{n} p\left(j|u\right)\delta\left(i, C_u\right) \geq 1$. Since $p\left(j|u\right)$ is a soft probability assignment, this corresponds to the case when at least one full point supports the mean.

- To also avoid the variances of clusters going to zero due to a small number of points being assigned to them, in the M-step you will need to implement a minimum variance for your clusters. We recommend a value of 0.25, though you are free to experiment with it if you wish. Note that this issue, as well as the thresholded mean update in the point above, are better dealt with through regularization; however, to keep things simple, we do not do regularization here.

- To debug your EM implementation, you may use the data files `test_incomplete.txt` and `test_complete.txt`. Compare your results to ours from `test_solutions.txt`.
  Correction note: The file `test_solutions.txt` has been updated on Aug 8. Please make sure to use the version in the latest netflix.tar.gz.

Implementing E-step (2)

1.0/1.0 point (graded)
In `em.py`, fill in the `estep` function so that it works with partially observed vectors where missing values are indicated with zeros, and perform the computations in the log domain to help with numerical stability.

Available Functions: You have access to the NumPy python library as `np`, to the `GaussianMixture` class and to typing annotation `typing.Tuple` as `Tuple`. You also have access to `scipy.special.logsumexp` as `logsumexp`

Hint: For this function, you will want to use `log(mixture.p[j] + 1e-16)` instead of `log(mixture.p[j])` to avoid numerical underflow

```
def estep(X: np.ndarray, mixture: GaussianMixture) -> Tuple[np.ndarray, float]:
    """E-step: Softly assigns each datapoint to a gaussian component

    Args:
        X: (n, d) array holding the data, with incomplete entries (set to 0)
        mixture: the current gaussian mixture

    Returns:
        np.ndarray: (n, K) array holding the soft counts
            for all components for all examples
        float: log-likelihood of the assignment

    """
    n, _ = X.shape

```

Press ESC then TAB or click outside of the code editor to exit

Correct

Test results

Hide output

CORRECT

Test: netflix

Output:

```
Input:
X: [[2. 4. 5. ... 4. 2. 4.]
 [1. 3. 4. ... 3. 0. 3.]
 [3. 5. 0. ... 3. 4. 0.]
 ...
 [5. 3. 0. ... 3. 5. 4.]
 [2. 3. 0. ... 5. 0. 0.]
 [3. 0. 0. ... 3. 0. 3.]]
K: 2
Mu: [[-0.80579745  0.67588981 -0.80780318  0.95291893 -0.0626976   0.95352218
   0.20969104  0.47852716 -0.92162442 -0.43438607 -0.75960688 -0.4077196
  -0.76254456 -0.36403364 -0.17147401 -0.87170501  0.38494424  0.13320291
  -0.46922102  0.04649611 -0.81211898  0.15189299  0.8585924  -0.3628621
   0.33482076 -0.73640428  0.43265441 -0.42118781 -0.63361728  0.17302587
  -0.95978491  0.65788006 -0.99060905  0.35563307 -0.45998405  0.47038804]
 [ 0.92437709 -0.50249371  0.15231467  0.18408386  0.14450381 -0.55383673
   0.90549802 -0.10574924  0.69281734  0.39895855 -0.4051261   0.62759564
  -0.20698852  0.76220639  0.16254575  0.76347072  0.38506318  0.45050856
   0.00264876  0.91216727  0.2879804  -0.1522899   0.21278643 -0.9616136
  -0.39685037  0.32034707 -0.41984479  0.23603086 -0.1424626  -0.72905187
  -0.40343535  0.13992982  0.18174552  0.1486505   0.30640164  0.30420654]]
Var: [11.52196427 10.0528577 ]
P: [0.4857973 0.5142027]
Output:
post:[[0.0441317  0.9558683 ]
 [0.26962778 0.73037222]
 [0.29128341 0.70871659]
 [0.13230341 0.86769659]
 [0.18954646 0.81045354]
 [0.13768804 0.86231196]
 [0.22805534 0.77194466]
 [0.22313851 0.77686149]
 [0.11155063 0.88844937]
 [0.12800874 0.87199126]
 [0.04440337 0.95559663]
 [0.30379257 0.69620743]
 [0.2527763  0.7472237 ]
 [0.2818586  0.7181414 ]
 [0.31792725 0.68207275]
 [0.26956845 0.73043155]
 [0.10092428 0.89907572]
 [0.20168715 0.79831285]
 [0.23715269 0.76284731]
 [0.12151621 0.87848379]
 [0.23212592 0.76787408]
 [0.15141622 0.84858378]
 [0.0433987  0.9566013 ]
 [0.11017693 0.88982307]
 [0.10948151 0.89051849]
 [0.05572526 0.94427474]
 [0.19283525 0.80716475]
 [0.1603763  0.8396237 ]
 [0.15684477 0.84315523]
 [0.03902817 0.96097183]
 [0.17248505 0.82751495]
 [0.1852236  0.8147764 ]
 [0.08303853 0.91696147]]
LL:-2347.129589
```

Test: output0

Output:

```
Input:
X: [[0.85794562 0.84725174]
 [0.6235637  0.38438171]
 [0.29753461 0.05671298]
 [0.27265629 0.47766512]
 [0.81216873 0.47997717]
 [0.3927848  0.83607876]
 [0.33739616 0.64817187]
 [0.36824154 0.95715516]
 [0.14035078 0.87008726]
 [0.47360805 0.80091075]
 [0.52047748 0.67887953]
 [0.72063265 0.58201979]
 [0.53737323 0.75861562]
 [0.10590761 0.47360042]
 [0.18633234 0.73691818]]
K: 6
Mu: [[0.6235637  0.38438171]
 [0.3927848  0.83607876]
 [0.81216873 0.47997717]
 [0.14035078 0.87008726]
 [0.36824154 0.95715516]
 [0.10590761 0.47360042]]
Var: [0.10038354 0.07227467 0.13240693 0.12411825 0.10497521 0.12220856]
P: [0.1680912  0.15835331 0.21384187 0.14223565 0.14295074 0.17452722]
Output:
post:[[0.17354324 0.19408461 0.38136556 0.0569083  0.16250611 0.03159219]
 [0.39379907 0.08689908 0.32081103 0.04067548 0.04920547 0.10860986]
 [0.35788286 0.01907566 0.18709725 0.04472511 0.01732312 0.37389601]
 [0.19268431 0.18091751 0.11938917 0.12743323 0.09677628 0.28279951]
 [0.36304946 0.07311615 0.43750366 0.02729566 0.04877955 0.05025552]
 [0.07858663 0.37039817 0.08705556 0.14917384 0.21407078 0.10071502]
 [0.13662023 0.29150288 0.10750309 0.13944117 0.14926196 0.17567066]
 [0.04532867 0.37841271 0.06233585 0.17307275 0.2613835  0.07946652]
 [0.03479877 0.30116079 0.03560306 0.24675099 0.22083886 0.16084754]
 [0.1084787  0.35703165 0.12209296 0.12356811 0.19771701 0.09111156]
 [0.18151437 0.29042408 0.1775779  0.09728296 0.14845737 0.10474333]
 [0.30076285 0.15240546 0.34401968 0.04831719 0.08817504 0.06631978]
 [0.14424702 0.32662602 0.16265301 0.10373169 0.17686354 0.08587872]
 [0.12020157 0.14175102 0.06966009 0.17178204 0.09140514 0.40520014]
 [0.06707408 0.29382796 0.05528713 0.20393925 0.17797873 0.20189285]]
LL:-5.592899
```

Test: output1

Output:

```
Input:
X: [[0.51048173 0.88065951 0.40148439 0.8244011 ]
 [0.24552216 0.01499308 0.22977642 0.3842103 ]
 [0.1381804  0.40847399 0.49885847 0.71721406]
 [0.78088514 0.72487674 0.66278747 0.87855436]
 [0.55499151 0.29798974 0.95371905 0.28823672]
 [0.71840591 0.33639791 0.11305392 0.31993637]
 [0.58349838 0.5956911  0.35595524 0.77428457]
 [0.72966102 0.86920654 0.01694379 0.62146977]
 [0.2994911  0.83032188 0.53017489 0.96366809]
 [0.84898977 0.28274182 0.79010911 0.39304625]
 [0.53370619 0.65802444 0.20953881 0.09547708]
 [0.1352663  0.03359603 0.2117697  0.86093549]
 [0.63900518 0.7882084  0.00980655 0.79854585]
 [0.6730584  0.64533222 0.51799599 0.56187788]
 [0.62717882 0.77827742 0.2782039  0.39993615]
 [0.86829312 0.48382955 0.52647043 0.02276892]
 [0.12109703 0.30775877 0.51761329 0.90921887]]
K: 4
Mu: [[0.24552216 0.01499308 0.22977642 0.3842103 ]
 [0.63900518 0.7882084  0.00980655 0.79854585]
 [0.51048173 0.88065951 0.40148439 0.8244011 ]
 [0.62717882 0.77827742 0.2782039  0.39993615]]
Var: [0.17209237 0.14078767 0.11686242 0.09997593]
P: [0.25991315 0.21765243 0.31606934 0.20636508]
Output:
post:[[0.01169278 0.16024756 0.63811528 0.18994439]
 [0.88432814 0.03496198 0.02691571 0.05379416]
 [0.30843384 0.10947319 0.43718381 0.14490916]
 [0.01533329 0.13203054 0.68540024 0.16723593]
 [0.50167939 0.03429033 0.19099417 0.27303611]
 [0.24754812 0.17044365 0.09806084 0.48394738]
 [0.04581753 0.19608358 0.49493737 0.26316152]
 [0.01389757 0.32635446 0.29371687 0.3660311 ]
 [0.0175261  0.1204496  0.77730389 0.08472041]
 [0.30686162 0.07511324 0.22424605 0.39377909]
 [0.10126591 0.09286732 0.10232582 0.70354095]
 [0.79378977 0.09149548 0.09111202 0.02360274]
 [0.01808066 0.38191577 0.37401387 0.2259897 ]
 [0.04125689 0.11788912 0.40547639 0.4353776 ]
 [0.02950736 0.13580527 0.25436274 0.58032463]
 [0.1556004  0.05861932 0.07983431 0.70594597]
 [0.37648253 0.116611   0.44347174 0.06343473]]
LL:-18.884699
```

Test: output2

Output:

```
Input:
X: [[0.48887885 0.02851583]
 [0.56854547 0.20662976]
 [0.16799477 0.51752097]
 [0.90532817 0.77648893]
 [0.84812979 0.45475271]
 [0.95399627 0.31685298]
 [0.61199078 0.72543565]
 [0.68219353 0.79633135]
 [0.57348016 0.55544709]
 [0.9063408  0.78498776]
 [0.16478768 0.32895619]
 [0.0685869  0.76794851]
 [0.0315623  0.45044255]
 [0.62397815 0.34637042]]
K: 5
Mu: [[-0.36903611  0.69340385]
 [ 0.94802773 -0.55528937]
 [ 0.24359843 -0.34254733]
 [-0.39534046 -0.46794313]
 [ 0.45627733  0.56920239]]
Var: [0.5084764  0.71834986 0.47809836 0.9872575  0.08089256]
P: [0.21548848 0.24189557 0.18488005 0.19983825 0.15789764]
Output:
post:[[0.11929132 0.20563789 0.2818886  0.10780441 0.28537778]
 [0.09435119 0.13566503 0.1685093  0.06698857 0.53448591]
 [0.16825394 0.05378286 0.09638093 0.05728383 0.62429844]
 [0.11860797 0.13615801 0.09187167 0.05457336 0.59878901]
 [0.08115254 0.14290169 0.11798126 0.05224555 0.60571896]
 [0.08758618 0.26345714 0.19232296 0.07828389 0.37834983]
 [0.08790773 0.05316746 0.05445099 0.03148722 0.7729866 ]
 [0.101206   0.06429812 0.05826487 0.0357941  0.74043691]
 [0.07495833 0.05586079 0.06409943 0.03195975 0.7731217 ]
 [0.12074838 0.13699614 0.09191884 0.0551108  0.59522583]
 [0.1776895  0.08061632 0.1515925  0.07915696 0.51094472]
 [0.29143209 0.0485012  0.08607928 0.06990288 0.50408455]
 [0.25628917 0.06965576 0.14343757 0.09035866 0.44025884]
 [0.07864555 0.09791622 0.11146946 0.04708999 0.66487878]]
LL:-22.109897
```

Test: output3

Output:

```
Input:
X: [[0.85794562 0.84725174]
 [0.6235637  0.38438171]
 [0.29753461 0.05671298]
 [0.         0.47766512]
 [0.         0.        ]
 [0.3927848  0.        ]
 [0.         0.64817187]
 [0.36824154 0.        ]
 [0.         0.87008726]
 [0.47360805 0.        ]
 [0.         0.        ]
 [0.         0.        ]
 [0.53737323 0.75861562]
 [0.10590761 0.        ]
 [0.18633234 0.        ]]
K: 6
Mu: [[0.6235637  0.38438171]
 [0.3927848  0.        ]
 [0.         0.        ]
 [0.         0.87008726]
 [0.36824154 0.        ]
 [0.10590761 0.        ]]
Var: [0.16865269 0.14023295 0.1637321  0.3077471  0.13718238 0.14220473]
P: [0.1680912  0.15835331 0.21384187 0.14223565 0.14295074 0.17452722]
Output:
post:[[0.65087662 0.05857439 0.02234959 0.20258382 0.0460844  0.01953118]
 [0.36462427 0.20175055 0.09281546 0.06127579 0.17543624 0.1040977 ]
 [0.10995174 0.22464491 0.20513252 0.02839796 0.21019956 0.22167331]
 [0.27996042 0.13156734 0.18479023 0.14012134 0.11793063 0.14563005]
 [0.1680912  0.15835331 0.21384187 0.14223565 0.14295074 0.17452722]
 [0.17188253 0.2079498  0.16224482 0.0981313  0.18938262 0.17040893]
 [0.33305679 0.09456056 0.14652199 0.23671559 0.08347925 0.10566582]
 [0.1634873  0.20447446 0.16926051 0.09967819 0.18702813 0.1760714 ]
 [0.34047752 0.04761128 0.08765585 0.42923507 0.04092387 0.0540964 ]
 [0.20164864 0.21756366 0.14029582 0.09378665 0.19519249 0.15151274]
 [0.1680912  0.15835331 0.21384187 0.14223565 0.14295074 0.17452722]
 [0.1680912  0.15835331 0.21384187 0.14223565 0.14295074 0.17452722]
 [0.47521046 0.09942182 0.06885849 0.20917529 0.08508798 0.06224596]
 [0.09128906 0.15565204 0.25208102 0.12427594 0.14824935 0.22845259]
 [0.11018277 0.17234878 0.22552021 0.1149792  0.16231678 0.21465225]]
LL:-8.829390
```

Test: output4

Output:

```
Input:
X: [[0.         0.98702305 0.        ]
 [0.03878139 0.49384448 0.7820454 ]
 [0.01583577 0.37313092 0.00937357]
 [0.12492591 0.31875794 0.97646793]
 [0.49284594 0.         0.        ]
 [0.         0.67998295 0.08640283]
 [0.         0.04676585 0.        ]
 [0.08601935 0.         0.30809923]
 [0.         0.         0.25312255]
 [0.60326212 0.28797558 0.        ]
 [0.         0.         0.        ]
 [0.         0.70350118 0.07305453]]
K: 3
Mu: [[0.12492591 0.31875794 0.97646793]
 [0.         0.70350118 0.07305453]
 [0.         0.98702305 0.        ]]
Var: [0.27882044 0.13985221 0.24665245]
P: [0.29092183 0.35849852 0.35057965]
Output:
post:[[0.14789921 0.43002153 0.42207925]
 [0.53981166 0.30231237 0.15787596]
 [0.05743606 0.72977937 0.21278456]
 [0.84275709 0.08802095 0.06922196]
 [0.34142453 0.31776976 0.34080571]
 [0.05097739 0.6534509  0.29557172]
 [0.59922521 0.25472238 0.14605241]
 [0.12722164 0.55814027 0.31463809]
 [0.12762502 0.505386   0.36698898]
 [0.52358016 0.28518686 0.19123298]
 [0.29092183 0.35849852 0.35057965]
 [0.04695597 0.650084   0.30296003]]
LL:-10.077751
```

```
Test: output5
```

Output:

```
Input:
X: [[0.90648012 0.99816002 0.13768454]
 [0.92950531 0.19200986 0.28557129]
 [0.         0.61078813 0.06522915]
 [0.87371365 0.99295672 0.9127793 ]
 [0.97548942 0.74477477 0.08488115]
 [0.56209216 0.08374213 0.38551777]
 [0.82880203 0.44056884 0.85701096]
 [0.23774516 0.85064934 0.10024266]
 [0.39181442 0.44763635 0.54246476]
 [0.45793145 0.83016375 0.93783038]
 [0.10680808 0.37806821 0.00726476]
 [0.54055095 0.36148986 0.83177797]
 [0.44364434 0.7694028  0.14101669]
 [0.01910687 0.69442013 0.73910354]
 [0.42275007 0.53663832 0.07744479]
 [0.         0.67327192 0.31881952]
 [0.28099472 0.4042464  0.92102663]
 [0.62297889 0.         0.47608691]
 [0.28456284 0.         0.        ]]
K: 8
Mu: [[-0.65524797 -0.46015273  0.88805118]
 [ 0.45180258 -0.67291488 -0.66320719]
 [ 0.5315099   0.45203706  0.0451484 ]
 [ 0.56613571  0.5355574   0.02038533]
 [ 0.40847458 -0.18143184  0.49460525]
 [ 0.4776481   0.10543376  0.5276616 ]
 [ 0.71248417  0.09990728 -0.34344878]
 [ 0.49106673  0.36453648  0.83323654]]
Var: [0.92359266 0.96794483 0.15097964 0.15727212 0.27362847 0.16667523
 0.37377302 0.17117464]
P: [0.14322866 0.11083623 0.12776976 0.12487704 0.11344219 0.11399134
 0.13670066 0.12915412]
```

```
Output:
post:[[0.00666831 0.01178137 0.32853384 0.44578778 0.02084131 0.03716047
  0.09414212 0.05508481]
 [0.0078168  0.01276342 0.24596604 0.20901218 0.09983314 0.21526298
  0.09456465 0.11478079]
 [0.02593187 0.01668103 0.34897876 0.3478211  0.0422044  0.07512775
  0.09260147 0.05065364]
 [0.02358267 0.01140786 0.07535356 0.09850656 0.0506973  0.10262185
  0.03913796 0.59869225]
 [0.00500262 0.01090391 0.34566567 0.41076444 0.02751182 0.05282541
  0.09956921 0.04775692]
 [0.01074047 0.00974002 0.188604   0.13680611 0.13050131 0.3078894
  0.05693837 0.15878032]
 [0.01224751 0.00670559 0.0710857  0.06552253 0.086416   0.23162923
  0.02840374 0.4979897 ]
 [0.01144853 0.00982745 0.37098475 0.39358529 0.03129523 0.05968677
  0.06200599 0.061166  ]
 [0.01120117 0.00600274 0.1884897  0.15616571 0.08016758 0.24135971
  0.03250545 0.28410793]
 [0.02233992 0.00643649 0.06355403 0.06729627 0.05672008 0.13955033
  0.0199661  0.62413676]
 [0.01604028 0.01486238 0.35499665 0.28657905 0.07437744 0.11933538
  0.08488985 0.04891896]
 [0.01336286 0.00545077 0.07078162 0.05801181 0.09436306 0.26714681
  0.02144305 0.46944002]
 [0.00690727 0.0071807  0.37232723 0.38659261 0.03043853 0.07169708
  0.05497603 0.06988054]
 [0.04067112 0.00978019 0.10240651 0.092748   0.08902899 0.18490947
  0.02744078 0.45301494]
 [0.0065779  0.00769183 0.38091867 0.34690468 0.04199197 0.09669648
  0.06108737 0.0581311 ]
 [0.03341068 0.01404392 0.28903423 0.28988736 0.05307833 0.1173957
  0.06744025 0.13570953]
 [0.0226616  0.00582798 0.04684079 0.03758072 0.09979668 0.24280535
```

```
  0.01642418 0.52806268]
 [0.02224088 0.02196408 0.16944373 0.1546288  0.14503447 0.24247282
  0.05609933 0.1881159 ]
 [0.0567504  0.06820591 0.16504415 0.15032444 0.12952371 0.15335748
  0.10750367 0.16929024]]
LL:-30.251463
```

Test: output6

Output:

```
Input:
X: [[0.         0.61510654 0.8403588 ]
 [0.68617182 0.         0.         ]
 [0.         0.         0.         ]
 [0.         0.04814391 0.         ]
 [0.63464793 0.63786054 0.         ]
 [0.         0.         0.         ]
 [0.         0.         0.7351466 ]
 [0.         0.85276462 0.         ]
 [0.         0.         0.         ]
 [0.         0.         0.         ]
 [0.         0.         0.04833832]
 [0.         0.         0.         ]
 [0.         0.         0.47222957]
 [0.         0.         0.         ]
 [0.         0.21868712 0.69472512]
 [0.         0.         0.67462913]
 [0.         0.         0.         ]]
K: 5
Mu: [[0.         0.         0.         ]
 [0.         0.         0.04833832]
 [0.         0.21868712 0.69472512]
 [0.         0.         0.         ]
 [0.         0.         0.         ]]
Var: [0.09501823 0.08922796 0.15708103 0.09501823 0.09501823]
P: [0.20556425 0.18647986 0.24404576 0.18856882 0.17534131]
Output:
post:[[0.00791577 0.00821619 0.96985477 0.00726132 0.00675196]
 [0.16597779 0.13230075 0.40789119 0.15225525 0.14157502]
 [0.20556425 0.18647986 0.24404576 0.18856882 0.17534131]
 [0.21940786 0.20523234 0.18694234 0.20126788 0.18714958]
 [0.08641648 0.0633163  0.69728423 0.07927183 0.07371116]
 [0.20556425 0.18647986 0.24404576 0.18856882 0.17534131]
 [0.05076835 0.05808408 0.80127244 0.04657097 0.04330416]
 [0.0654045  0.04776382 0.77104618 0.05999705 0.05578845]
 [0.20556425 0.18647986 0.24404576 0.18856882 0.17534131]
 [0.20556425 0.18647986 0.24404576 0.18856882 0.17534131]
 [0.25219068 0.23900459 0.0623519  0.23134032 0.21511252]
 [0.20556425 0.18647986 0.24404576 0.18856882 0.17534131]
 [0.15561224 0.17207864 0.39682897 0.14274669 0.13273346]
 [0.20556425 0.18647986 0.24404576 0.18856882 0.17534131]
 [0.0639483  0.07411652 0.7487276  0.05866124 0.05454634]
 [0.07130635 0.08128429 0.72117582 0.06541096 0.06082258]
 [0.20556425 0.18647986 0.24404576 0.18856882 0.17534131]]
LL:-13.030951
```

Hide output

Submit

You have used 6 of 20 attempts

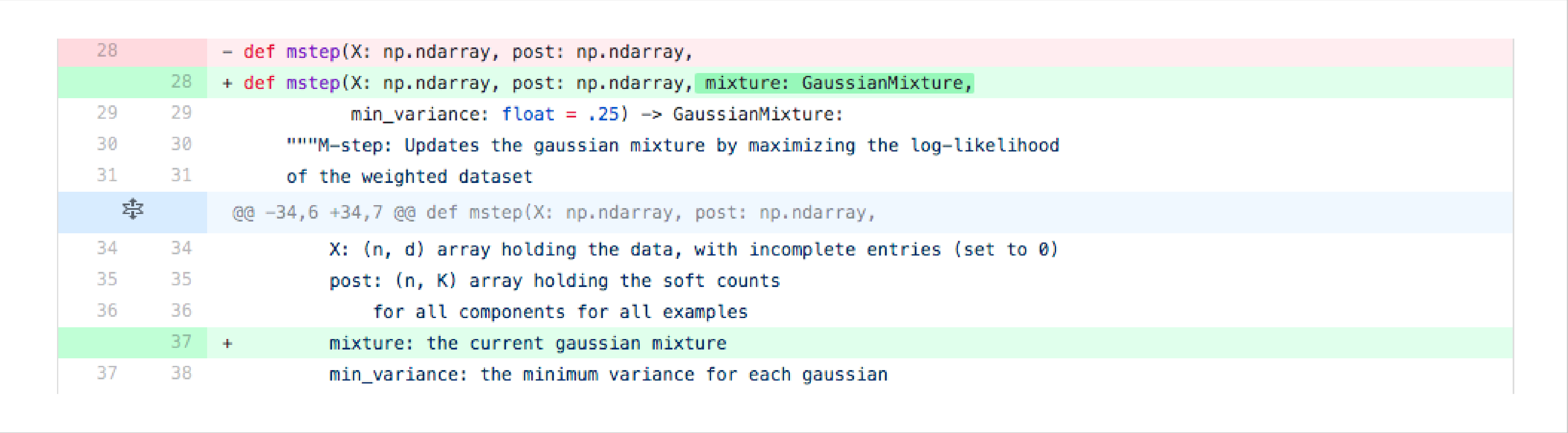
Implementing M-step (2)

1/1 point (graded)

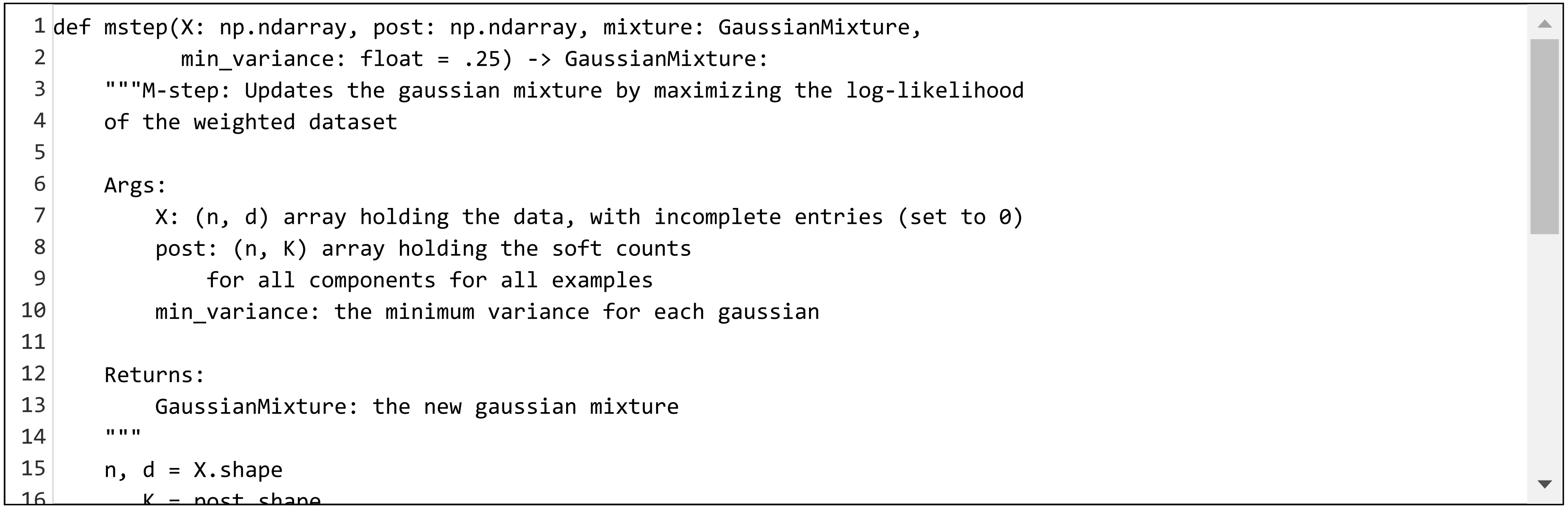
In `em.py`, fill in the `mstep` function so that it works with partially observed vectors where missing values are indicated with zeros, and perform the computations in the log domain to help with numerical stability.

Available Functions: You have access to the NumPy python library as `np`, to the `GaussianMixture` class and to typing annotation `typing.Tuple` as `Tuple`. You also have access to `scipy.misc.logsumexp` as `logsumexp`

Correction Note (Aug 8): The boilerplate code for this function was changed on August 8th. Make sure you have the latest version of netflix.tar.gz, or correct the file `em.py` as follows:



```
- def mstep(X: np.ndarray, post: np.ndarray,
+ def mstep(X: np.ndarray, post: np.ndarray, mixture: GaussianMixture,
            min_variance: float = .25) -> GaussianMixture:
      """M-step: Updates the gaussian mixture by maximizing the log-likelihood
      of the weighted dataset
@@ -34,6 +34,7 @@ def mstep(X: np.ndarray, post: np.ndarray,
          X: (n, d) array holding the data, with incomplete entries (set to 0)
          post: (n, K) array holding the soft counts
              for all components for all examples
+         mixture: the current gaussian mixture
          min_variance: the minimum variance for each gaussian
```



```
def mstep(X: np.ndarray, post: np.ndarray, mixture: GaussianMixture,
          min_variance: float = .25) -> GaussianMixture:
    """M-step: Updates the gaussian mixture by maximizing the log-likelihood
    of the weighted dataset

    Args:
        X: (n, d) array holding the data, with incomplete entries (set to 0)
        post: (n, K) array holding the soft counts
            for all components for all examples
        min_variance: the minimum variance for each gaussian

    Returns:
        GaussianMixture: the new gaussian mixture
    """
    n, d = X.shape
       K = post.shape
```

Press ESC then TAB or click outside of the code editor to exit

Correct

Test results

Hide output

CORRECT

Test: netflix

Output:

```
Input:
X: [[2. 4. 5. ... 4. 2. 4.]
 [1. 3. 4. ... 3. 0. 3.]
 [3. 5. 0. ... 3. 4. 0.]
 ...
 [5. 3. 0. ... 3. 5. 4.]
 [2. 3. 0. ... 5. 0. 0.]
 [3. 0. 0. ... 3. 0. 3.]]
Mu: [[3.53162741 3.6748473  3.9759755  3.45370958 3.8008092  3.7733913
  3.85593062 3.89326771 3.91092966 3.9239507  4.03777538 3.83217966
  3.54926155 3.59040243 3.58355266 3.43232952 3.81496564 3.66189157
  3.60734369 3.85686273 3.82554477 3.96335373 3.64985243 3.89208143
  3.99971012 3.71984898 3.30438117 3.5298734  3.81919097 2.96419876
  3.70156056 3.88719678 3.62173912 3.86828322 3.95426438 3.88642264]
 [3.31054003 3.76038323 3.63550775 3.04156335 3.87735873 3.80912846
  3.73186432 4.01106405 3.85168698 3.44669611 3.9627543  3.8344115
  3.63389477 3.50817988 3.45981734 3.47578916 3.97785621 3.55067041
  3.80692371 3.99041455 3.89986666 4.12572231 3.40788236 3.72183376
  4.00029395 3.5064034  3.41221426 3.33566486 3.80191824 3.03462194
  3.82139677 4.09015403 3.59644227 3.88112641 4.04803705 4.02136671]]
K: 2
post:[[0.43418691 0.56581309]
 [0.52521691 0.47478309]
 [0.39610605 0.60389395]
 [0.32917129 0.67082871]
 [0.7153587  0.2846413 ]
 [0.59951013 0.40048987]
 [0.38030858 0.61969142]
 [0.44912526 0.55087474]
 [0.02370719 0.97629281]
 [0.47213411 0.52786589]
 [0.55047309 0.44952691]
 [0.37155893 0.62844107]
 [0.15599464 0.84400536]
 [0.13175585 0.86824415]
 [0.55722649 0.44277351]
 [0.25467687 0.74532313]
 [0.44520528 0.55479472]
 [0.02952397 0.97047603]
 [0.49803167 0.50196833]
 [0.58056498 0.41943502]
 [0.45133699 0.54866301]
 [0.92053184 0.07946816]
 [0.49855274 0.50144726]
 [0.62003262 0.37996738]
 [0.46445322 0.53554678]
 [0.56522377 0.43477623]
 [0.90641685 0.09358315]
 [0.56424771 0.43575229]
 [0.72055205 0.27944795]
 [0.65609528 0.34390472]
 [0.59020864 0.40979136]
 [0.82607833 0.17392167]
 [0.3477441  0.6522559 ]]
Output:
Mu: [[3.53162741 3.6748473  3.9759755  3.45370958 3.8008092  3.7733913
  3.85593062 3.89326771 3.91092966 3.9239507  4.03777538 3.83217966
  3.54926155 3.59040243 3.58355266 3.43232952 3.81496564 3.66189157
  3.60734369 3.85686273 3.82554477 3.96335373 3.64985243 3.89208143
  3.99971012 3.71984898 3.30438117 3.5298734  3.81919097 2.96419876
  3.70156056 3.88719678 3.62173912 3.86828322 3.95426438 3.88642264]
 [3.31054003 3.76038323 3.63550775 3.04156335 3.87735873 3.80912846
  3.73186432 4.01106405 3.85168698 3.44669611 3.9627543  3.8344115
  3.63389477 3.50817988 3.45981734 3.47578916 3.97785621 3.55067041
  3.80692371 3.99041455 3.89986666 4.12572231 3.40788236 3.72183376
  4.00029395 3.5064034  3.41221426 3.33566486 3.80191824 3.03462194
  3.82139677 4.09015403 3.59644227 3.88112641 4.04803705 4.02136671]]
```

```
Var: [0.96833845 0.98603321]
P: [0.4857973 0.5142027]
```

Test: output0

Output:

```
Input:
X: [[0.85794562 0.84725174]
 [0.6235637  0.38438171]
 [0.29753461 0.05671298]
 [0.27265629 0.47766512]
 [0.81216873 0.47997717]
 [0.3927848  0.83607876]
 [0.33739616 0.64817187]
 [0.36824154 0.95715516]
 [0.14035078 0.87008726]
 [0.47360805 0.80091075]
 [0.52047748 0.67887953]
 [0.72063265 0.58201979]
 [0.53737323 0.75861562]
 [0.10590761 0.47360042]
 [0.18633234 0.73691818]]
K: 6
post:[[0.15765074 0.20544344 0.17314824 0.15652173 0.12169798 0.18553787]
 [0.1094766  0.22310587 0.24109142 0.0959303  0.19807563 0.13232018]
 [0.22679645 0.36955206 0.02836173 0.03478709 0.00807236 0.33243031]
 [0.16670188 0.18637975 0.20964608 0.17120102 0.09886116 0.16721011]
 [0.04250305 0.22996176 0.05151538 0.33947585 0.18753121 0.14901275]
 [0.09799086 0.28677458 0.16895715 0.21054678 0.0069597  0.22877093]
 [0.16764519 0.16897033 0.25848053 0.18674186 0.09846462 0.11969746]
 [0.28655211 0.02473762 0.27387452 0.27546459 0.08641467 0.05295649]
 [0.11353057 0.13090863 0.20522811 0.15786368 0.35574052 0.03672849]
 [0.10510461 0.08116927 0.3286373  0.12745369 0.23464272 0.12299241]
 [0.09757735 0.06774952 0.40286261 0.08481828 0.1206645  0.22632773]
 [0.24899344 0.02944918 0.25413459 0.02914503 0.29614373 0.14213403]
 [0.35350682 0.21890411 0.26755234 0.01418274 0.10235276 0.04350123]
 [0.15555757 0.06236572 0.16703133 0.21760554 0.03369562 0.36374421]
 [0.1917808  0.08982788 0.17710673 0.03179658 0.19494387 0.31454414]]
Output:
```

```
Mu: [[0.43216722 0.64675402]
 [0.46139681 0.57129172]
 [0.44658753 0.68978041]
 [0.44913747 0.66937822]
 [0.47080526 0.68008664]
 [0.40532311 0.57364425]]
Var: [0.25 0.25 0.25 0.25 0.25 0.25]
P: [0.1680912  0.15835331 0.21384187 0.14223565 0.14295074 0.17452722]
```

Test: output1

Output:

```
Input:
X: [[9.41995544e-01 9.56255069e-01 7.79478257e-01 5.03139435e-01]
 [2.64580162e-02 1.85308100e-01 4.06890361e-01 5.03050977e-01]
 [9.27804780e-01 7.05796516e-01 4.68583265e-01 9.75763136e-01]
 [2.80704425e-01 5.03729309e-01 8.26921053e-01 1.26142597e-04]
 [6.02372886e-01 3.20045302e-01 7.19656132e-01 8.35817014e-01]
 [8.16461473e-01 4.15420687e-01 6.00908597e-01 3.28429598e-01]
 [7.10235389e-01 4.26127432e-01 9.35371239e-01 2.17460410e-01]
 [6.74172704e-01 2.71845904e-01 6.65797628e-01 7.21895428e-01]
 [2.48483665e-01 3.62987466e-01 7.08598791e-01 9.55245263e-01]
 [3.47168111e-01 3.97425130e-01 2.83222653e-01 2.70250083e-01]
 [8.84791303e-02 2.29879069e-01 4.77760087e-01 5.79624169e-01]
 [3.91641448e-01 8.15017742e-01 8.52424164e-01 9.26888118e-01]
 [6.69903699e-01 9.76212934e-01 5.98228984e-01 3.62710510e-01]
 [3.73392184e-01 4.91584032e-01 3.10999262e-01 4.66870006e-01]
 [6.10114673e-02 2.53910297e-01 8.43239609e-01 2.20800378e-01]
 [9.89021822e-01 9.61608139e-01 8.97271972e-01 3.98854569e-03]
 [5.82040701e-02 1.41426450e-01 1.17630815e-01 5.45998041e-01]
 [2.55139525e-01 8.58160291e-01 1.63086068e-01 1.84228206e-01]
 [9.23698918e-01 2.77283027e-01 2.70550499e-02 7.57837465e-02]]
K: 8
post:[[7.61584771e-02 9.38062734e-02 2.30466773e-01 2.33955480e-01
  1.90705614e-01 1.23097102e-01 6.47316609e-03 4.53371144e-02]
 [8.52604446e-02 1.05410091e-01 1.94413669e-01 1.47893709e-01
  9.81876725e-02 2.04462938e-01 5.88192456e-02 1.05552230e-01]
 [1.82268842e-01 2.78041839e-05 1.32774233e-01 7.05439611e-02
  1.58625650e-01 1.84229678e-01 1.79963355e-01 9.15664770e-02]
 [1.44291319e-01 7.88631419e-02 1.70543077e-01 1.02322532e-01
  2.24603127e-01 5.22170087e-02 1.61883636e-01 6.52761573e-02]
 [1.51056676e-01 1.63784188e-01 5.63761641e-02 8.23548744e-02
  1.60767436e-01 2.16726777e-01 7.87657670e-02 9.01681178e-02]
 [9.03169637e-02 8.61801369e-02 2.82151386e-02 7.33062108e-02
  1.52353069e-01 1.84836540e-01 1.24890668e-01 2.59901273e-01]
 [1.62324933e-01 1.76504911e-01 1.27568032e-01 1.85897709e-01
  1.13919201e-01 6.90700260e-02 7.11041096e-02 9.36110782e-02]
 [7.57156949e-02 1.13663893e-01 1.48538154e-02 6.18168494e-02
  2.05294612e-01 5.37559283e-02 2.40786663e-01 2.34112544e-01]
 [3.11788806e-01 1.38596093e-03 2.02250578e-02 4.91436103e-02
  4.08749772e-02 1.89726285e-01 8.86572304e-02 2.98198073e-01]
 [6.02132575e-02 6.80191788e-02 3.41040295e-01 1.02376092e-01
  9.98903648e-03 2.79802333e-02 3.36417305e-02 3.56740176e-01]
 [4.15692805e-02 1.96653516e-01 1.91899773e-01 2.46802889e-01
  2.92621256e-02 1.87876218e-02 1.76751940e-01 9.82728536e-02]
 [7.29905015e-02 2.15541393e-01 5.56108193e-02 6.98759719e-02
  1.09872660e-01 1.52622931e-01 2.18092069e-01 1.05393654e-01]
 [1.13807040e-01 1.89666461e-01 8.73076926e-02 1.96536345e-01
  2.46770021e-01 1.22771577e-01 1.33724129e-02 2.97684501e-02]
 [9.89428247e-02 1.68094540e-01 1.69279300e-01 1.64915422e-01
  4.90344976e-02 1.70635018e-01 1.46252026e-02 1.64473195e-01]
 [1.67296799e-01 2.11943870e-01 2.05211084e-01 9.01665301e-02
  1.33704231e-01 1.05970153e-01 7.26147627e-02 1.30925703e-02]
 [2.04143156e-01 1.02979189e-01 1.66124469e-01 1.18118927e-01
  2.65484343e-02 9.01406280e-02 2.00256568e-01 9.16886279e-02]
 [9.49521016e-02 1.55316671e-01 5.74239807e-02 1.24275569e-01
  1.36545523e-01 2.05210081e-01 9.22668487e-02 1.34009224e-01]
 [1.54871297e-01 1.68170501e-01 1.77102675e-01 8.64950416e-02
  1.17146316e-01 5.68475255e-02 1.50090050e-01 8.92765945e-02]
 [1.99846160e-01 7.27298164e-02 1.12491901e-01 5.84121644e-02
  1.65453202e-01 1.46231570e-01 4.93266312e-02 1.95508555e-01]]
Output:
Mu: [[0.52264789 0.51128949 0.58244058 0.45483975]
 [0.43644772 0.5060929  0.58010718 0.43700242]
 [0.45848182 0.52309614 0.53739036 0.37087466]
 [0.48446415 0.52750496 0.57435985 0.43184724]
 [0.54900488 0.52751691 0.58137732 0.45102904]
 [0.50313366 0.48267067 0.54377741 0.5316927 ]
 [0.50345368 0.51624612 0.60941299 0.49855754]
 [0.49631555 0.43344867 0.5022292  0.48655729]]
Var: [0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25]
P: [0.13093761 0.12467061 0.13362779 0.11922157 0.12471881 0.12501682
 0.10696748 0.13483931]
```

Test: output2

Output:

```
Input:
X: [[0.5733608  0.777866   0.21832627 0.31598716]
 [0.40656884 0.12863997 0.17884917 0.01941066]
 [0.70299964 0.52397865 0.06475103 0.89897763]
 [0.64216815 0.75546345 0.48178057 0.33449363]
 [0.47952615 0.99086612 0.87006496 0.39647243]
 [0.42721727 0.61854707 0.31550533 0.63308097]
 [0.34278254 0.01438643 0.61013304 0.51270627]
 [0.67347221 0.45293618 0.14845276 0.14816382]
 [0.35472294 0.71806163 0.92797545 0.54017645]
 [0.79387491 0.47207855 0.1878827  0.17397495]
 [0.45809409 0.85284807 0.70113644 0.08137186]
 [0.9317442  0.66381735 0.22212494 0.62809731]
 [0.14536082 0.93028856 0.84614347 0.58897132]
 [0.51647322 0.69722323 0.22120176 0.99414949]
 [0.04618445 0.55543267 0.82867931 0.11106542]
 [0.12661646 0.99378599 0.52844925 0.36749583]
 [0.16162908 0.71730726 0.97117231 0.45500442]
 [0.35812339 0.04941717 0.31173839 0.21790621]
 [0.00942554 0.7588793  0.87220105 0.14626169]]
K: 9
post:[[0.04868641 0.12415421 0.11061081 0.11390858 0.12997212 0.11551315
  0.0763783  0.12665907 0.15411735]
 [0.08817827 0.22809675 0.04318663 0.14350484 0.12130935 0.05240019
  0.14502884 0.0907368  0.08755834]
 [0.02706332 0.18239863 0.09041306 0.15699109 0.16119508 0.22117883
  0.1252733  0.0156473  0.0198394 ]
 [0.05366813 0.08610823 0.00945619 0.25127316 0.12469348 0.01670058
  0.13379883 0.29123408 0.0330673 ]
 [0.01781341 0.23711668 0.18788459 0.1194587  0.08171602 0.05535127
  0.03688928 0.07894611 0.18482394]
 [0.26339868 0.17744064 0.08967867 0.00853491 0.00598243 0.09217854
  0.03798972 0.17653218 0.14826423]
 [0.0529766  0.15711225 0.07527218 0.14977765 0.00386846 0.23658784
  0.02734035 0.13459249 0.16247217]
 [0.09358886 0.10455276 0.0856146  0.17198708 0.05718472 0.01416195
  0.13972487 0.14749316 0.18569199]
 [0.19310088 0.04757709 0.03390468 0.07699783 0.03429588 0.18600031
  0.23325498 0.08254298 0.11232536]
 [0.05538141 0.10216456 0.0109098  0.19742967 0.09146251 0.20584765
  0.20112319 0.10947338 0.02620783]
 [0.21996064 0.09604093 0.04483528 0.22450716 0.03580668 0.05214392
  0.16649076 0.01492156 0.14529308]
 [0.00578983 0.1791824  0.19716392 0.09839051 0.11611846 0.14684752
  0.03191412 0.06222802 0.16236521]
 [0.10504031 0.13292785 0.03627746 0.05959326 0.15954509 0.12114194
  0.10072147 0.15550984 0.12924278]
 [0.04344782 0.04518892 0.10954993 0.09486911 0.14552519 0.00213849
  0.23045246 0.21242787 0.1164002 ]
 [0.10395639 0.14038924 0.14035021 0.07953989 0.06729972 0.08060089
  0.12696025 0.08302297 0.17788044]
 [0.01263325 0.18317212 0.21251416 0.12006728 0.07069142 0.0900843
  0.06433294 0.13489599 0.11160855]
 [0.22902852 0.1974601  0.01828543 0.0846496  0.09221425 0.06173937
  0.07395456 0.05262826 0.19003992]
 [0.00837798 0.0053719  0.14543174 0.23356472 0.15959738 0.03496203
  0.1683721  0.01123124 0.2330909 ]
 [0.06827071 0.06134422 0.17847073 0.03510192 0.18046278 0.13715933
  0.10779405 0.0554861  0.17591016]]
Output:
Mu: [[0.36850431 0.64999064 0.60332085 0.38273866]
 [0.43287571 0.63017412 0.5016101  0.40548497]
 [0.40629279 0.6491047  0.49660251 0.40693754]
 [0.48602017 0.56695479 0.44605726 0.35201311]
 [0.43087687 0.62531725 0.46839554 0.4144754 ]
 [0.44626388 0.58149028 0.50395507 0.44192488]
 [0.44146286 0.59526716 0.4796192  0.38808552]
 [0.44865942 0.64478601 0.46985415 0.43571034]
 [0.38629877 0.60440098 0.54770631 0.3723846 ]]
Var: [0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25]
P: [0.08896639 0.13093682 0.09577948 0.12737616 0.09678637 0.10119674
 0.11725234 0.10716892 0.1345368 ]
```

Test: output3

Output:

```
Input:
X: [[0.85794562 0.84725174]
 [0.6235637  0.38438171]
 [0.29753461 0.05671298]
 [0.         0.47766512]
 [0.         0.        ]
 [0.3927848  0.        ]
 [0.         0.64817187]
 [0.36824154 0.        ]
 [0.         0.87008726]
 [0.47360805 0.        ]
 [0.         0.        ]
 [0.         0.        ]
 [0.53737323 0.75861562]
 [0.10590761 0.        ]
 [0.18633234 0.        ]]
K: 6
post:[[0.15765074 0.20544344 0.17314824 0.15652173 0.12169798 0.18553787]
 [0.1094766  0.22310587 0.24109142 0.0959303  0.19807563 0.13232018]
 [0.22679645 0.36955206 0.02836173 0.03478709 0.00807236 0.33243031]
 [0.16670188 0.18637975 0.20964608 0.17120102 0.09886116 0.16721011]
 [0.04250305 0.22996176 0.05151538 0.33947585 0.18753121 0.14901275]
 [0.09799086 0.28677458 0.16895715 0.21054678 0.0069597  0.22877093]
 [0.16764519 0.16897033 0.25848053 0.18674186 0.09846462 0.11969746]
 [0.28655211 0.02473762 0.27387452 0.27546459 0.08641467 0.05295649]
 [0.11353057 0.13090863 0.20522811 0.15786368 0.35574052 0.03672849]
 [0.10510461 0.08116927 0.3286373  0.12745369 0.23464272 0.12299241]
 [0.09757735 0.06774952 0.40286261 0.08481828 0.1206645  0.22632773]
 [0.24899344 0.02944918 0.25413459 0.02914503 0.29614373 0.14213403]
 [0.35350682 0.21890411 0.26755234 0.01418274 0.10235276 0.04350123]
 [0.15555757 0.06236572 0.16703133 0.21760554 0.03369562 0.36374421]
 [0.1917808  0.08982788 0.17710673 0.03179658 0.19494387 0.31454414]]
Output:
```

```
Mu: [[0.41970858 0.57419602]
 [0.46514641 0.50509648]
 [0.45167656 0.64343368]
 [0.41703438 0.87008726]
 [0.36824154 0.        ]
 [0.35384957 0.44162885]]
Var: [0.25       0.25       0.25       0.25       0.28690463 0.25      ]
P: [0.1680912  0.15835331 0.21384187 0.14223565 0.14295074 0.17452722]
```

Test: output4

Output:

```
Input:
X: [[0.41302022 0.80568642 0.97341576 0.        ]
 [0.         0.21428313 0.         0.22828444]
 [0.         0.         0.         0.80299306]
 [0.         0.         0.06689605 0.32115162]
 [0.64175749 0.         0.26444098 0.48990335]
 [0.         0.36123146 0.         0.07620241]
 [0.78224446 0.         0.         0.96473494]
 [0.         0.         0.         0.30750159]
 [0.38740023 0.10287016 0.         0.71796623]
 [0.96279109 0.08311093 0.         0.88729796]
 [0.         0.97495569 0.         0.        ]
 [0.66535744 0.64184406 0.         0.        ]
 [0.38493484 0.11285293 0.         0.29756222]
 [0.80760164 0.33047493 0.0024602  0.77602101]
 [0.12086067 0.         0.15266383 0.33157782]
 [0.         0.         0.         0.82081681]
 [0.28440147 0.0457341  0.         0.35984439]
 [0.48495255 0.         0.51449559 0.63105734]]
K: 7
post:[[0.35065711 0.04493986 0.00059463 0.13922617 0.3265943  0.04116011
  0.09682782]
 [0.20387035 0.20451042 0.10719777 0.045066   0.21122652 0.06916243
  0.15896652]
 [0.105958   0.26959489 0.04778842 0.20113784 0.07021378 0.02839512
  0.27691195]
 [0.21887585 0.21079891 0.03900651 0.19300232 0.12226255 0.11634771
  0.09970615]
 [0.03597559 0.08483176 0.22673067 0.24554393 0.1933648  0.11047057
  0.10308268]
 [0.17044687 0.03556631 0.24655812 0.1153721  0.01359442 0.18721792
  0.23124426]
 [0.25331535 0.01443969 0.14552245 0.02140191 0.35422402 0.07631039
```

```
  0.13478619]
 [0.11145125 0.24862975 0.2020313  0.07136485 0.16351809 0.05597653
  0.14702823]
 [0.05985935 0.22331274 0.17889739 0.07211828 0.18708517 0.09282607
  0.18590101]
 [0.28162042 0.13738448 0.06248091 0.06315076 0.17143181 0.24516013
  0.03877148]
 [0.0577445  0.19838012 0.1371933  0.1307459  0.16500181 0.14713037
  0.16380401]
 [0.13349193 0.34336347 0.15376867 0.00103006 0.0848264  0.00677737
  0.27674209]
 [0.20484581 0.11547    0.19544345 0.03502777 0.19380606 0.03109788
  0.22430903]
 [0.28404999 0.04970828 0.1312017  0.37884097 0.01746614 0.02819648
  0.11053644]
 [0.31118556 0.08349529 0.36732842 0.02473862 0.02931326 0.07109962
  0.11283923]
 [0.17691962 0.12908237 0.22098529 0.04161721 0.18248234 0.13365374
  0.11525943]
 [0.08195526 0.04955203 0.36742564 0.20797379 0.03646487 0.17189624
  0.08473217]
 [0.05360773 0.14997772 0.17002329 0.17540373 0.13312816 0.16276318
  0.15509618]]
Output:
Mu: [[ 0.55808479  0.37191182  0.35179687  0.5470557 ]
 [ 0.54999135  0.40849803  0.91137329  0.53171147]
 [ 0.45207913  0.28516784 -0.85661235  0.47784552]
 [ 0.57743589  0.37269509  0.26653949  0.55269904]
 [ 0.57719668  0.4201962   0.2404478   0.59912946]
 [ 0.57359434  0.30767505 -0.54052991  0.53672485]
 [ 0.52087962  0.39508027 -0.0405426   0.51157935]]
Var: [0.25 0.25 0.25 0.25 0.25 0.25 0.25]
P: [0.17199059 0.14405767 0.16667655 0.12015346 0.14755581 0.09864677
 0.15091916]
```

```
Test: output5
```

Output:

```
Input:
X: [[0.78689629 0.09400376 0.07744925 0.        ]
 [0.84079045 0.4413037  0.34782583 0.14677624]
 [0.39705116 0.86795612 0.21630449 0.41440401]
 [0.24964489 0.73835282 0.93373003 0.00644357]
 [0.64454426 0.         0.03164945 0.224879  ]
 [0.89041102 0.         0.62166799 0.99524137]
 [0.59496131 0.69751894 0.50125511 0.65798358]
 [0.52407928 0.24983097 0.0099372  0.99739169]
 [0.90895019 0.78479587 0.06117337 0.49803869]
 [0.31006729 0.17199605 0.83315662 0.82403888]
 [0.         0.11495315 0.07054325 0.39648556]
 [0.38386803 0.36397939 0.80064718 0.02655044]
 [0.41323823 0.890323   0.47395928 0.95986581]
 [0.57531001 0.20879088 0.93477956 0.51629613]
 [0.57357962 0.63435392 0.37584822 0.14855098]
 [0.95561678 0.65705703 0.87472619 0.61143665]
 [0.24036884 0.53054995 0.52499823 0.82750204]
 [0.55069006 0.2189513  0.26781941 0.4605273 ]
 [0.07062603 0.25411614 0.88964287 0.        ]]
K: 6
post:[[0.12443987 0.16975377 0.33456555 0.03390002 0.30775791 0.02958288]
 [0.07785043 0.21443926 0.12455146 0.17407861 0.15874749 0.25033276]
 [0.07439055 0.22662227 0.15304639 0.12493978 0.20035378 0.22064723]
 [0.19210278 0.26850417 0.24710636 0.01728444 0.01270171 0.26230054]
 [0.22901328 0.21576447 0.1120658  0.20208684 0.17871678 0.06235283]
 [0.19281146 0.15746068 0.22536131 0.15869169 0.07408305 0.19159181]
 [0.12331965 0.25118093 0.05147742 0.21741487 0.18228175 0.17432536]
 [0.0201343  0.30411796 0.20236472 0.23585085 0.09271092 0.14482126]
 [0.09401818 0.10737709 0.17353382 0.06306259 0.4385446  0.12346371]
 [0.229483   0.23934817 0.17434564 0.24758315 0.05117378 0.05806627]
 [0.25587881 0.17792199 0.17459726 0.12092943 0.01309839 0.25757412]
 [0.12455063 0.090569   0.26253966 0.24401981 0.20543559 0.07288531]
 [0.16763959 0.23244022 0.17283876 0.10839533 0.26456946 0.05411664]
 [0.12567134 0.00909786 0.3179707  0.09531057 0.31184361 0.14010592]
 [0.22252515 0.16361006 0.17858399 0.1713604  0.20031749 0.0636029 ]
 [0.32023799 0.06386017 0.15973307 0.05196554 0.35614523 0.04805799]
 [0.11826704 0.09214597 0.01837655 0.41334099 0.15720801 0.20066144]
 [0.16794876 0.21789147 0.0778413  0.15209377 0.18427645 0.19994825]
 [0.1045187  0.04036922 0.25761923 0.2217911  0.27170411 0.10399764]]
Output:
Mu: [[0.57293679 0.46585475 0.50484492 0.49661664]
 [0.54386928 0.48689451 0.39809972 0.52512429]
 [0.55103249 0.42216926 0.50565753 0.48830177]
 [0.48602136 0.43618804 0.47219551 0.56157927]
 [0.60087479 0.50349836 0.46165964 0.51307149]
 [0.53585435 0.47629885 0.45077006 0.48650723]]
Var: [0.25 0.25 0.25 0.25 0.25 0.25]
P: [0.15604219 0.17065656 0.17992205 0.16074209 0.19271948 0.13991762]
```

Test: output6

Output:

```
Input:
X: [[0.         0.5976806 ]
 [0.         0.03417229]
 [0.         0.        ]
 [0.         0.        ]
 [0.         0.        ]
 [0.         0.        ]
 [0.         0.        ]
 [0.1837871  0.        ]
 [0.         0.        ]
 [0.         0.        ]
 [0.89607341 0.36542401]
 [0.03380226 0.        ]
 [0.         0.        ]
 [0.         0.        ]
 [0.20355604 0.        ]
 [0.96932775 0.        ]
 [0.         0.        ]
 [0.65053523 0.        ]]
K: 6
post:[[0.22890554 0.25668829 0.08569813 0.32316526 0.08706585 0.01847692]
 [0.14288769 0.00211009 0.27266944 0.19777315 0.27577711 0.10878251]
 [0.25987749 0.10538583 0.17916883 0.25871358 0.04919885 0.14765541]
 [0.07908285 0.15167075 0.21185722 0.17537447 0.27135468 0.11066003]
 [0.01278211 0.2011026  0.20299328 0.31028414 0.24008473 0.03275313]
 [0.07936955 0.06262639 0.37795541 0.09906593 0.001849   0.37913372]
 [0.16797105 0.23609457 0.06250913 0.21695686 0.0799127  0.23655569]
 [0.17056019 0.02024888 0.27329603 0.00206656 0.27838756 0.25544078]
 [0.03933287 0.21765246 0.0809767  0.27943472 0.31442761 0.06817565]
 [0.00975892 0.15618372 0.17478696 0.18535875 0.23178719 0.24212446]
 [0.26521739 0.23249264 0.24765606 0.10497894 0.09896903 0.05068595]
 [0.24034748 0.09194291 0.22718543 0.15257227 0.16827689 0.11967502]
 [0.35491349 0.10413661 0.07994663 0.14529151 0.22155359 0.09415816]
 [0.21108932 0.05331763 0.20425903 0.17590148 0.10733065 0.24810189]
 [0.10825589 0.19985029 0.16212442 0.14752162 0.22928306 0.15296473]
 [0.1860168  0.0148476  0.2299486  0.39470553 0.08224313 0.09223834]
 [0.00085679 0.17652878 0.32074619 0.05455891 0.30141801 0.14589133]
 [0.13797226 0.18104714 0.31727677 0.01373004 0.16091519 0.1890586 ]]
Output:
Mu: [[0.51357264 0.        ]
 [0.20355604 0.        ]
 [0.50917929 0.        ]
 [0.         0.        ]
 [0.36992289 0.        ]
 [0.03380226 0.        ]]
Var: [0.25       0.25       0.25       0.41416433 0.25       0.25      ]
P: [0.1497332  0.13688484 0.20616968 0.17985854 0.1777686  0.14958513]
```

Hide output

Submit You have used 5 of 20 attempts

✓ Correct (1/1 point)

Implementing run

1.0/1.0 point (graded)

In `em.py`, fill in the `run` function so that it runs the EM algorithm. As before, the convergence criteria that you should use is that the improvement in the log-likelihood is less than or equal to 10^{-6} multiplied by the absolute value of the new log-likelihood.

Available Functions: You have access to the NumPy python library as `np`, to the `GaussianMixture` class and to typing annotation `typing.Tuple` as `Tuple`. You also have access to the `estep` and `mstep` functions you have just implemented

Correction note (Aug 8): Since the `mstep` function in previous problem has been defined differently since Aug 8, you will need to modify `run` function accordingly. Note that the grader will accept as correct a `run` function that works with either the earlier or current version of `mstep`.

```
    Returns:
        GaussianMixture: the new gaussian mixture
        np.ndarray: (n, K) array holding the soft counts
            for all components for all examples
        float: log-likelihood of the current assignment
    """
    prev_log_likelihood = None
    log_likelihood = None
    while (prev_log_likelihood is None or log_likelihood - prev_log_likelihood > 1e-6*np.abs(log_likelihood)):
        prev_log_likelihood = log_likelihood
        post, log_likelihood = estep(X, mixture)
        mixture = mstep(X, post)
        #print(log_likelihood)

    return mixture, post, log_likelihood

```

Press ESC then TAB or click outside of the code editor to exit

Correct

Test results

Hide output

CORRECT

Test: starting fixed

Output:

```
Cost = -521.0000
Cost = -401.2500
Cost = -356.3438
Cost = -336.6973
Cost = -327.4880
Cost = -323.0272
Cost = -320.8317
Cost = -319.7425
Cost = -319.2000
Cost = -318.9293
Cost = -318.7941
Cost = -318.7265
Cost = -318.6928
Cost = -318.6759
Cost = -318.6674
Cost = -318.6632
Cost = -318.6611
Cost = -318.6601
Cost = -318.6595
Cost = -318.6593
```

Test: starting random

Output:

```
Cost = -1058.9247
Cost = -1058.9244
```

Hide output

Submit You have used 0 of 20 attempts

Discussion

Hide Discussion

Topic: Unit 4 Unsupervised Learning (2 weeks) :Project 4: Collaborative Filtering via Gaussian Mixtures / 7. Implementing EM for matrix completion

Add a Post

All Posts

Implementing M-step (2)

discussion posted 3 days ago by Mark_B2 (Community TA)

May be it's just me, but I didn't understand the phrase "*Instead, we should only update the mean when* $\sum_{u=1}^{n} p(j|u)\,\delta(i, C_u) \geq 1$". I assumed the update is only possible when μ has previous value, otherwise it's assignment. Grader's output suggest the previous value should be zero.

This post is visible to everyone.

Add a Response

3 responses

RommelAlbertoRodriguezPerez
2 days ago

My very same doubt EDIT: If more people have this doubt, do no forget to press "+" and maybe add some [staff] tag

Checking out the output it indeed seems that when they mean not to update, they just mean "set the new mu_j to 0"

posted a day ago by **RommelAlbertoRodriguezPerez**

@Mark_B2, I also got confused first. I think what they meant is at each M-Step, for each $\mu[k]$, you assign values of for all dimensions to 0 ("unknown") first and then update only for the dimensions that have enough information.

posted a day ago by **OlgaSkv**

@OlgaSkv I agree, that was possibly their thought. But this is implementation dependent. For example I did not initialize μ's, just get it as direct matrix manipulation. Therefore I thought, it was worth clarification.

posted a day ago by **Mark_B2** (Community TA)

Add a comment

karenechu (Staff)
about 18 hours ago

Thank you for your comments. This is now fixed. See correction note in the the problem.

Add a comment

Hryhorchuk
about 4 hours ago

I implemented M-step, and it works fine on most of the test cases, but it sometimes returns $\mu = 0$ where grader returns $\mu > 0$. Looks like I misinterpret the update rule. Referring to one of the tests:

The grader returns mu: [[0. 0.] [**0.17077953** 0.] [0. 0.] [0.61917624 0.]]

My implementation returns mu: [[0. 0.] [**0.** , 0.], [0. 0.], [0.61917624, 0.]]

There is a difference in mu[1,0].

My interpretation of the update rule for mu[1,0] is the following. We have non-zero X[4,0], X[6,0], X[10,0] and X[13,0]. We recalculate mu[1,0] if post[4,0]+post[6,0]+post[10,0]+post[13,0] >=1, and assign mu[1,0]=0 otherwise.

post[4,0]+post[6,0]+post[10,0]+post[13,0] = 0.23252945 + 0.29152496 + 0.00090738 + 0.34208213 <1, hence I set mu[1,0]=0.

What am I doing wrong? Will appreciate any thoughts.

The arguments from test that I'm referring:

```
X = np.array([[0., 0.],
[0., 0.],
[0., 0.],
[0., 0.],
[**0.17077953**, 0.],
[0., 0.],
[**0.65314958**, 0.01464121],
[0., 0.],
[0., 0.],
[0., 0.],
[**0.98563941**, 0.],
[0., 0.47802869],
[0., 0.],
[**0.0823144**, 0.],
[0., 0.],
[0., 0.],
[0., 0.],
[0., 0.]])
post = np.array([[0.40381723, 0.27523853, 0.25332294, 0.06762131],
[0.04432647, 0.18039234, 0.31115808, 0.46412311],
[0.27246332, 0.15833883, 0.31201891, 0.25717893],
[0.02566033, 0.47051072, 0.06054002, 0.44328893],
[**0.23252945**, 0.3908681, 0.0010023, 0.37560015],
[0.07663558, 0.33399099, 0.27854979, 0.31082364],
[**0.29152496**, 0.12584098, 0.03860641, 0.54402765],
[0.01281825, 0.26340296, 0.37786832, 0.34591048],
[0.36697866, 0.32284731, 0.02818523, 0.28198881],
[0.06327807, 0.06041602, 0.53939154, 0.33691438],
[**0.00090738**, 0.15582033, 0.16019182, 0.68308047],
[0.33832996, 0.25748032, 0.31452121, 0.08966851],
[0.16762585, 0.11403558, 0.27313747, 0.4452011],
[**0.34208213**, 0.15387307, 0.31705369, 0.18699111],
[0.41280524, 0.0678067, 0.438963, 0.08042506],
[0.15417507, 0.22448936, 0.21914827, 0.4021873],
[0.5783029, 0.05763114, 0.09396838, 0.27009758],
[0.25958613, 0.25692869, 0.29887059, 0.1846146]])
```

•••

OK, I finally got it. The update algorithm that is used by the grader has been changed, and now it is different from the one described by Mark_B2 in the original post. I should have been more mindful about why "mixture" has been added as an argument to the function.

Now, you have to compare your estimated mu with previous one, and make an **update** if necessary. If estimated mu[j,i]=0 for some j and i, you have to look up for previous_mu[j,i].

I spent about 5 hours trying to understand what is going wrong, so I hope this helps someone to avoid the same mistake.

This have been my hardest 0.475 points in this course to date)

posted about an hour ago by **Hryhorchuk**

Add a comment

Showing all responses

Add a response:

Preview

Submit