edX

3. Expectation–maximization algorithm

Recall the Gaussian mixture model presented in class:

$$P\left(x|\theta\right) = \sum_{j=1}^{K} \pi_j N\left(x; \mu^{(j)}, \sigma_j^2 I\right),$$

where θ denotes all the parameters in the mixture (means $\mu^{(j)}$, mixing proportions π_j, and variances σ_j^2). The goal of the EM algorithm is to estimate these unknown parameters by maximizing the log-likelihood of the observed data $x^{(1)}, \ldots, x^{(n)}$. Starting with some initial guess of the unknown parameters, the algorithm iterates between E- and M-steps. The E-Step softly assigns each data point $x^{(i)}$ to mixture components. The M-step takes these soft-assignments as given and finds a new setting of the parameters by maximizing the log-likelihood of the weighted dataset (expected complete log-likelihood).

Implement the EM algorithm for the Gaussian mixture model desribed above. To this end, complete the functions `estep`, `mstep` and `run` in `naive_em.py`. In our notation,

- x: an (n, d) Numpy array of n data points, each with d features

- `K`: number of mixture components

- `mu`: (K, d) Numpy array where the jth row is the mean vector $\mu^{(j)}$

- `p`: $(K,)$ Numpy array of mixing proportions π_j, $j = 1, \ldots, K$

- `var`: $(K,)$ Numpy array of variances σ_j^2, $j = 1, \ldots, K$

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. In slightly more algebraic notation:

$$\text{new log-likelihood} - \text{old log-likelihood} \leq 10^{-6} \cdot |\text{new log-likelihood}|$$

Your code will output updated versions of a `GaussianMixture` (with means `mu`, variances `var` and mixing proportions `p`) as well as an (n, K) Numpy array `post`, where $\text{post}[i, j]$ is the posterior probability $p\left(j|x^{(i)}\right)$, and `LL` which is the log-likelihood of the weighted dataset.

Here are a few points to check to make sure that your implementation is indeed correct:

1. Make sure that all your functions return objects with the right dimension.

2. EM should monotonically increase the log-likelihood of the data. Initialize and run the EM algorithm on the toy dataset as you did earlier with K-means. You should check that the `LL` values that the algorithm returns after each run are indeed always monotonically increasing (non-decreasing).

3. Using $K = 3$ and a seed of 0, on the toy dataset, you should get a log likelihood of -1388.0818.

4. As a runtime guideline, in your testing on the toy dataset, calls of `run` using the values of K that we are testing should run in on the order of seconds (i.e. if each call isn't fairly quick, that may be an indication that something is wrong).

5. Try plotting the solutions obtained with your EM implementation. Do they make sense?

Implementing E-step

1/1 point (graded)

Write a function `estep` that performs the E-step of the EM algorithm

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

```
1 def estep(X: np.ndarray, mixture: GaussianMixture) -> Tuple[np.ndarray, float]:
2     """E-step: Softly assigns each datapoint to a gaussian component
3
4     Args:
5         X: (n, d) array holding the data
6         mixture: the current gaussian mixture
7
8     Returns:
9         np.ndarray: (n, K) array holding the soft counts
10             for all components for all examples
11         float: log-likelihood of the assignment
12     """
13     n, _ = X.shape
14     K, d = mixture.mu.shape
15     post = np.zeros((n, K))
16
```

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

Correct

Test results

CORRECT

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: [[0.83175331 0.68587044 0.32144683 0.94968377]
 [0.81367278 0.82635961 0.6010079  0.84433016]
 [0.32651691 0.20008727 0.3337401  0.78560828]
 [0.74593502 0.11093618 0.40498703 0.98053332]
 [0.80504665 0.47631084 0.02881411 0.64948477]
 [0.46022002 0.54428103 0.17120706 0.96573027]
 [0.14803813 0.94506788 0.16330805 0.26253357]
 [0.49491903 0.37818026 0.81396891 0.03199006]
 [0.89207027 0.97015169 0.44173703 0.04791278]
 [0.93793998 0.34255353 0.83928703 0.76153207]
 [0.18051169 0.01838981 0.30297445 0.70733705]
 [0.33405579 0.64406413 0.76835968 0.58036825]
 [0.77114317 0.68161258 0.89577837 0.26550793]]
K: 9
Mu: [[ 0.04441322 -0.04376179  0.0101338  -0.63448439]
 [-0.10792904  0.53572393  0.7482083  -0.98992122]
 [-0.43647738 -0.43825745  0.048467   -0.09286427]
 [-0.54195061  0.72930514  0.83473203 -0.62780362]
 [ 0.99238616  0.10449139  0.2295279   0.15189772]
 [-0.94393282  0.18806236  0.65205231  0.37203685]
 [ 0.6694693   0.88051619  0.03945951 -0.77215409]
 [-0.12239552 -0.70391247  0.03546318  0.27303161]
 [ 0.00474923 -0.86486349 -0.57883556 -0.1702664 ]]
Var: [0.67736778 0.86289415 0.74870738 0.83150465 0.23416358 0.72817616
 0.63695409 0.6659064  1.07908606]
P: [0.13722065 0.08589696 0.10676954 0.12556337 0.09772688 0.12354121
 0.10114632 0.10580856 0.11632651]
Output:
post:[[0.0577906  0.02151204 0.03990667 0.03453227 0.64064387 0.05153845
  0.06730799 0.05835175 0.02841636]
 [0.06508201 0.03614393 0.04004185 0.06020363 0.56057361 0.06925071
  0.09406216 0.05020016 0.02444194]
```

```
 [0.08969862 0.02245496 0.08851707 0.04012572 0.44721299 0.10148661
  0.03496673 0.13566127 0.03987604]
 [0.05179254 0.0130774  0.05080305 0.01960769 0.65976416 0.04736252
  0.02455196 0.09945162 0.03358906]
 [0.05321301 0.01247003 0.03028841 0.01674734 0.7486862  0.02200968
  0.04974543 0.04444143 0.02239847]
 [0.08635769 0.02458087 0.07821091 0.04632357 0.44428989 0.10556041
  0.05803911 0.11284906 0.04378847]
 [0.16159262 0.06879921 0.07948643 0.12969417 0.17148824 0.12052421
  0.17677967 0.05981439 0.03182107]
 [0.11589453 0.05760205 0.05252733 0.07804425 0.48073352 0.05712784
  0.08616465 0.05187765 0.02002818]
 [0.08491033 0.04901378 0.02312516 0.05881209 0.52123086 0.02276542
  0.20641932 0.019053   0.01467005]
 [0.04733898 0.02247743 0.03239008 0.03107624 0.71552121 0.03952108
  0.03998484 0.0517408  0.01994934]
 [0.10819392 0.02340219 0.11877603 0.04185633 0.32512742 0.12081873
  0.02898872 0.18327012 0.04956653]
 [0.10268063 0.05768072 0.07208161 0.11122239 0.3245684  0.14893958
  0.08058532 0.07637943 0.02586193]
 [0.07915378 0.05394215 0.03307151 0.07424782 0.55579061 0.0465825
  0.10825605 0.03324803 0.01570756]]
LL:-56.608674
```

Test: output2

Output:

```
Input:
X: [[0.03324135 0.92056043]
 [0.71801815 0.08894104]
 [0.72369287 0.69507826]
 [0.16358272 0.02185078]
 [0.20021071 0.49450292]
 [0.6562261  0.22888875]
 [0.35537809 0.07942668]
 [0.92374749 0.09946256]
 [0.0960033  0.78444218]
 [0.41114935 0.06829204]
 [0.34056114 0.32086192]
 [0.47693271 0.66142014]
 [0.45123772 0.99746843]
 [0.57819465 0.99731327]
 [0.39022627 0.83555327]
 [0.96119972 0.87454655]
 [0.10003248 0.30847221]]
K: 8
Mu: [[0.20021071 0.49450292]
 [0.03324135 0.92056043]
 [0.71801815 0.08894104]
 [0.34056114 0.32086192]
 [0.10003248 0.30847221]
 [0.6562261  0.22888875]
 [0.72369287 0.69507826]
 [0.0960033  0.78444218]]
Var: [0.13067895 0.2746292  0.22146461 0.12184699 0.17838113 0.15901117
 0.15838532 0.20253656]
P: [0.14472957 0.14684438 0.09587244 0.12890331 0.14673814 0.09583634
 0.12628561 0.11479021]
Output:
post:[[0.22207838 0.23890672 0.01408059 0.07333824 0.12700162 0.017656
```

```
  0.06736916 0.23956929]
 [0.09182437 0.02805198 0.18786633 0.20518514 0.10692598 0.24299752
  0.10847845 0.02867022]
 [0.13093001 0.08051261 0.07430419 0.12828624 0.07155884 0.11802944
  0.3137143  0.08266438]
 [0.18436302 0.04686233 0.08420581 0.25355161 0.25412333 0.09658461
  0.02785518 0.05245411]
 [0.26251569 0.08656459 0.03863862 0.20436884 0.17204922 0.05950899
  0.07007848 0.10627556]
 [0.12545834 0.0362985  0.1350115  0.22321508 0.11164331 0.19816696
  0.1301256  0.04008072]
 [0.16798304 0.03925094 0.10341118 0.26753881 0.19018269 0.13589595
  0.05049133 0.04524607]
 [0.05292208 0.02379623 0.25309596 0.13787429 0.06991308 0.2937631
  0.14753567 0.02109959]
 [0.25685077 0.17116261 0.0202209  0.11427043 0.14536377 0.02838495
  0.07474545 0.18900111]
 [0.15479615 0.03648041 0.11610171 0.26490157 0.1771503  0.15277754
  0.05627292 0.04151939]
 [0.21670768 0.05538941 0.06581292 0.250499   0.16555297 0.1015856
  0.07634015 0.06811228]
 [0.20672177 0.0920302  0.05042339 0.16950804 0.10844248 0.08419679
  0.18246494 0.10621239]
 [0.1647706  0.1918162  0.02850207 0.07662465 0.07669438 0.04108047
  0.23556142 0.18495021]
 [0.13570395 0.17154995 0.03580295 0.071464   0.06382609 0.05142572
  0.31126646 0.15896087]
 [0.21181682 0.14338951 0.03306424 0.12101498 0.10218233 0.05197098
  0.18072309 0.15583806]
 [0.05841479 0.09333152 0.07901746 0.05201242 0.03522094 0.10192104
  0.50652988 0.07355195]
 [0.25130091 0.07217287 0.04412646 0.22443855 0.22142125 0.06012088
  0.03921858 0.0872005 ]]
LL:-14.850918
```

Test: output3

Output:

```
Input:
X: [[0.55612077 0.52289855 0.06046727]
 [0.19778    0.72366386 0.26867883]
 [0.72023066 0.33001366 0.20715301]
 [0.09564113 0.93015775 0.08649504]
 [0.61448735 0.17432515 0.36664573]
 [0.35335295 0.96884826 0.42267006]
 [0.64549216 0.936743   0.09607771]
 [0.13611427 0.22409364 0.15676573]
 [0.78965112 0.69173407 0.88595491]
 [0.88187672 0.97519618 0.83133236]
 [0.68989864 0.37294501 0.32872468]
 [0.64259458 0.09909546 0.19922167]]
K: 9
Mu: [[-0.91037876 -0.83442477  0.5445984 ]
 [ 0.199299   -0.34858579 -0.40950329]
 [-0.18174838 -0.65989093 -0.82378493]
 [ 0.2437633  -0.05076761 -0.83553319]
 [ 0.43448922  0.32071395  0.95492008]
 [-0.9385935  -0.33190548  0.82425818]
 [-0.72649126  0.10339698  0.7220826 ]
 [-0.78266826 -0.50705346  0.26824891]
 [-0.65884531 -0.46065992  0.19968823]]
Var: [1.44886645 0.5811562  1.19793203 0.68682719 0.23525379 1.15363282
 0.72970524 1.0443014  0.91262804]
P: [0.11591778 0.1287818  0.11621271 0.10766599 0.10533909 0.11861991
 0.10903304 0.09896477 0.09946489]
Output:
post:[[0.03131168 0.22709079 0.05757738 0.156589   0.3034537  0.04167158
  0.0753912  0.04695236 0.05996231]
 [0.03423242 0.13586295 0.04258874 0.09379131 0.39823559 0.05511693
  0.1213619  0.05290942 0.06590074]
 [0.02848529 0.1989133  0.04793358 0.1164815  0.42122534 0.03628681
```

```
  0.05978643 0.03998506 0.0509027 ]
 [0.04413551 0.1692321  0.06268317 0.14752474 0.19603714 0.07070358
  0.15490499 0.06957386 0.08520491]
 [0.02731853 0.15536121 0.03708928 0.0758586  0.51970938 0.03622225
  0.06140712 0.03822815 0.04880548]
 [0.03130093 0.08895422 0.03428429 0.07009841 0.51506666 0.05247275
  0.11172667 0.04394189 0.05215419]
 [0.03524054 0.18355938 0.0624643  0.17006953 0.30242365 0.04937657
  0.08850316 0.04909786 0.05926501]
 [0.03907997 0.22055632 0.05461453 0.11531897 0.25729111 0.05566101
  0.11115372 0.06337543 0.08294895]
 [0.01641211 0.03118899 0.0128164  0.01834836 0.8121828  0.02589106
  0.04365599 0.01850444 0.02099985]
 [0.02210503 0.03678575 0.01852544 0.02775059 0.75419011 0.03483904
  0.05620007 0.02370068 0.02590328]
 [0.02528676 0.14617969 0.03675343 0.08281765 0.53724233 0.03401713
  0.05857751 0.03500387 0.04412164]
 [0.03216305 0.23514818 0.05308731 0.11901169 0.35280962 0.03950428
  0.06286109 0.04593554 0.05947923]]
LL:-41.409041
```

Test: output4

Output:

```
Input:
X: [[0.83869151 0.25190057]
 [0.86402822 0.7732177 ]
 [0.18220298 0.13382016]
 [0.78110846 0.44651542]
 [0.93989485 0.39806437]
 [0.93810407 0.78762466]
 [0.04438112 0.98739929]
 [0.57324553 0.88825487]
 [0.98397011 0.7542914 ]
 [0.74128059 0.96319333]
 [0.64354157 0.68869758]
 [0.3728567  0.7456542 ]
 [0.89868371 0.02630872]
 [0.50039258 0.00483401]
 [0.55159022 0.2933089 ]
 [0.82298114 0.7851806 ]
 [0.78673243 0.17725026]
 [0.65426703 0.32563177]]
K: 3
Mu: [[0.64354157 0.68869758]
 [0.82298114 0.7851806 ]
 [0.93810407 0.78762466]]
Var: [0.09829708 0.12962052 0.15412959]
P: [0.31327108 0.25898718 0.42774173]
Output:
post:[[0.36567804 0.24495989 0.38936207]
 [0.33762501 0.2791076  0.3832674 ]
 [0.54469383 0.1927908  0.26251537]
 [0.41469415 0.24617736 0.33912849]
 [0.32467375 0.260215   0.41511125]
 [0.29442423 0.28660354 0.41897224]
 [0.48415746 0.24450487 0.27133767]
```

```
        [0.43838224 0.26042733 0.30119044]
        [0.27549199 0.28697028 0.43753773]
        [0.34447858 0.28686142 0.36866   ]
        [0.46065773 0.24607503 0.29326725]
        [0.53357133 0.22459533 0.24183334]
        [0.27951359 0.24117719 0.47930922]
        [0.44495483 0.2135356  0.34150957]
        [0.50240315 0.21558209 0.28201476]
        [0.35656529 0.27609454 0.36734017]
        [0.37832613 0.23817479 0.38349907]
        [0.46663185 0.22707788 0.30629027]]
    LL:-13.444727
```

Test: output5

Output:

```
Input:
X: [[0.25748733 0.05967311 0.28446856]
 [0.00517448 0.11449595 0.62496237]
 [0.10784829 0.4861792  0.21025798]
 [0.41049844 0.17696355 0.16136806]
 [0.5083836  0.27865588 0.60799757]
 [0.73901511 0.9326943  0.84216631]
 [0.0668398  0.87619025 0.14671773]
 [0.4906614  0.6410448  0.3419088 ]
 [0.50985693 0.78654764 0.44637756]
 [0.33891412 0.59186802 0.70879268]
 [0.97488104 0.1348835  0.81942076]
 [0.60492881 0.4024021  0.52424426]
 [0.38358588 0.96654423 0.9326852 ]
 [0.46025335 0.34221709 0.8605098 ]
 [0.1400456  0.70190433 0.4238915 ]
 [0.02182739 0.92143239 0.17461265]]
K: 6
Mu: [[0.33891412 0.59186802 0.70879268]
 [0.25748733 0.05967311 0.28446856]
 [0.97488104 0.1348835  0.81942076]
 [0.0668398  0.87619025 0.14671773]
 [0.46025335 0.34221709 0.8605098 ]
 [0.00517448 0.11449595 0.62496237]]
Var: [0.09354658 0.17167848 0.28099987 0.19411104 0.13331484 0.18499621]
P: [0.18941473 0.19198168 0.16805203 0.18486862 0.15143401 0.11424893]
Output:
post:[[0.10159037 0.51046125 0.05080864 0.0636585  0.1076959  0.16578535]
 [0.18969585 0.28923384 0.03607569 0.04860382 0.17449018 0.26190063]
 [0.23688034 0.27941962 0.02343049 0.27458334 0.0705659  0.1151203 ]
 [0.11401976 0.51077596 0.06508194 0.09970469 0.09804978 0.11236787]
 [0.36710643 0.16611536 0.07863046 0.03476406 0.27579427 0.07758941]
 [0.52222924 0.02290861 0.12492351 0.07315578 0.23844529 0.01833757]
```

```
       [0.16114155 0.09933679 0.01318314 0.65156882 0.02664806 0.04812163]
       [0.45768832 0.13865938 0.05107695 0.17401475 0.13155147 0.04700914]
       [0.5416142  0.07545087 0.04764767 0.17149938 0.13074884 0.03303904]
       [0.60067762 0.06231424 0.03363316 0.0583303  0.1939471  0.05109758]
       [0.0860944  0.09489191 0.41543966 0.00720335 0.3588212  0.03754947]
       [0.41029415 0.14999552 0.08767124 0.05238449 0.24412084 0.05553376]
       [0.63282577 0.01847753 0.04624536 0.08905029 0.18502256 0.02837849]
       [0.42323252 0.07879381 0.07098881 0.02045062 0.33954352 0.06699071]
       [0.47386271 0.1072896  0.02030305 0.23549109 0.09284507 0.07020849]
       [0.15131074 0.0818924  0.01142809 0.68586783 0.02365635 0.04584459]]
      LL:-17.628757
```

Hide output

Submit   You have used 3 of 20 attempts

✔ Correct (1/1 point)

Implementing M-step

1/1 point (graded)

Write a function `mstep` that performs the M-step of the EM algorithm

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

```
1 def mstep(X: np.ndarray, post: np.ndarray) -> GaussianMixture:
```

```
 2     """M-step: Updates the gaussian mixture by maximizing the log-likelihood
 3     of the weighted dataset
 4
 5     Args:
 6         X: (n, d) array holding the data
 7         post: (n, K) array holding the soft counts
 8             for all components for all examples
 9
10     Returns:
11         GaussianMixture: the new gaussian mixture
12     """
13     n, d = X.shape
14     _, K = post.shape
15
```

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

Correct

Test results

Hide output

CORRECT

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: [0.05218451 0.06230449 0.03538519 0.05174859 0.04524244 0.05831186]
P: [0.1680912  0.15835331 0.21384187 0.14223565 0.14295074 0.17452722]
```

Test: output1

Output:

```
Input:
X: [[0.3525339  0.35110968]
 [0.78744858 0.18143875]
 [0.31049287 0.98669636]
 [0.04966694 0.00793196]
 [0.39550754 0.17534554]
 [0.37087813 0.49396609]
 [0.72793558 0.06011484]
 [0.22740509 0.58985766]
 [0.84971916 0.52751794]
 [0.34061858 0.84626854]
 [0.9943659  0.35026881]
 [0.11170356 0.89183797]
 [0.33343344 0.8932389 ]
 [0.58736723 0.39683075]
 [0.70207427 0.51288305]
 [0.9436354  0.26415728]
 [0.67968175 0.0333847 ]
 [0.92564812 0.24467019]]
K: 3
post:[[0.11397883 0.63180888 0.25421229]
 [0.20613076 0.57218917 0.22168007]
 [0.26599279 0.59655331 0.13745391]
 [0.23053157 0.73259222 0.0368762 ]
 [0.0137045  0.68334099 0.30295451]
 [0.23284959 0.3101283  0.45702211]
 [0.0685165  0.25918726 0.67229625]
 [0.49463943 0.30707931 0.19828126]
 [0.38626468 0.45386116 0.15987416]
 [0.08354948 0.6670566  0.24939393]
 [0.47577573 0.31285591 0.21136836]
 [0.32524629 0.23760066 0.43715306]
 [0.27031402 0.69552317 0.03416281]
```

```
    [0.55465544 0.14660825 0.29873631]
    [0.22796188 0.39843261 0.37360551]
    [0.41530326 0.30418895 0.28050779]
    [0.13438614 0.55471847 0.3108954 ]
    [0.43823394 0.12597324 0.43579282]]
Output:
Mu: [[0.58355732 0.46295448]
 [0.48407832 0.44165773]
 [0.57979007 0.39286609]]
Var: [0.08512154 0.09181033 0.07569101]
P: [0.27433527 0.44387214 0.2817926 ]
```

Test: output2

Output:

```
Input:
X: [[0.14972802 0.83456216 0.89660305]
 [0.01838092 0.89956816 0.03199891]
 [0.43858367 0.98869508 0.52337896]
 [0.34838309 0.17501539 0.03009453]
 [0.23871661 0.82960472 0.25123644]
 [0.04620164 0.27188834 0.99683785]
 [0.03780173 0.18305858 0.28411085]
 [0.67876193 0.28912662 0.37693038]
 [0.01240371 0.0167197  0.98733743]
 [0.06577768 0.32277455 0.75759055]
 [0.05695046 0.33284362 0.89929077]
 [0.84634572 0.60520882 0.95193443]
 [0.84836608 0.29814493 0.67144696]
 [0.61617074 0.33031098 0.59326911]
 [0.34757181 0.37784335 0.90635567]
 [0.34137107 0.51904763 0.66892681]
 [0.30990286 0.98387507 0.2757363 ]
 [0.21087923 0.22947317 0.74854284]
 [0.08047794 0.62177415 0.45636925]]
K: 8
post:[[5.18835707e-02 1.67550607e-01 4.12854585e-02 2.30119122e-01
  2.47226052e-01 5.06828780e-03 2.48043640e-01 8.82326129e-03]
 [1.22767558e-01 2.76753762e-01 1.46503305e-01 9.75187735e-02
  4.89899950e-02 8.42400615e-03 6.68211286e-02 2.32221471e-01]
 [9.13621155e-02 1.68012228e-02 9.88721771e-02 3.62500023e-01
  1.37465964e-02 6.65692402e-02 1.03316892e-01 2.46831732e-01]
 [1.02213258e-01 1.33254012e-01 4.38501161e-03 5.91081980e-03
  3.49047680e-01 2.32540008e-02 1.14108618e-01 2.67826599e-01]
 [1.17688489e-02 6.87823486e-02 1.85839017e-01 1.74897889e-01
  1.25066792e-01 1.96717864e-01 1.75315398e-01 6.16118419e-02]
 [1.60466648e-01 1.47256387e-01 7.89398091e-02 1.41783209e-01
  8.30649122e-02 9.02993971e-02 2.16606619e-01 8.15830196e-02]
```

```
[1.31524871e-01 1.69503735e-01 7.85283095e-02 2.49310531e-01
 6.98706217e-02 5.34360654e-02 5.81477068e-02 1.89678159e-01]
[1.80614699e-02 1.39543271e-01 1.02421848e-01 7.90528970e-02
 7.88605202e-02 1.54514676e-01 2.03298195e-01 2.24247123e-01]
[1.87819521e-01 1.70641145e-01 1.58304061e-01 1.33134599e-01
 6.34372755e-02 4.05842241e-02 1.12016485e-01 1.34062690e-01]
[1.48005208e-01 8.55935931e-02 1.23248457e-01 2.39323789e-02
 1.79906452e-01 1.92774493e-01 1.66398530e-01 8.01408884e-02]
[4.75564259e-02 1.63785780e-01 1.51926067e-01 4.86255272e-02
 1.06357346e-01 2.00193619e-01 8.23771005e-02 1.99178134e-01]
[7.15802896e-02 3.67586793e-02 1.64149889e-01 1.63516270e-01
 1.29924578e-01 1.69607636e-02 2.45520080e-01 1.71589451e-01]
[1.54712650e-01 4.01556728e-02 4.77254473e-02 1.80044652e-01
 1.88186734e-01 1.70624247e-01 4.31310843e-02 1.75419514e-01]
[6.60696772e-03 1.11133782e-01 8.29409868e-02 2.58534391e-01
 2.34137398e-01 2.51147288e-01 1.55890687e-02 3.99101177e-02]
[1.59155164e-01 1.10219417e-01 2.13431939e-01 6.44574936e-02
 1.87678116e-01 1.73036563e-01 1.11233233e-03 9.09089748e-02]
[1.75790440e-01 1.44024636e-01 8.86879323e-02 3.38606504e-02
 1.54426111e-01 1.48139957e-01 9.35509360e-02 1.61519339e-01]
[1.80481423e-01 2.72896952e-03 1.76171102e-01 1.84054288e-01
 5.14572719e-02 1.60558049e-01 7.96260262e-02 1.64922870e-01]
[1.71064707e-01 1.96404209e-01 1.23185968e-01 9.36947062e-02
 1.67479242e-01 1.86050134e-04 2.15132670e-01 3.28524470e-02]
[1.31594501e-01 1.11494774e-02 1.77610192e-01 1.68828506e-01
 1.78938662e-01 6.62394460e-02 1.09422882e-01 1.56216334e-01]]
Output:
Mu: [[0.25501683 0.44861573 0.62342829]
 [0.2233891  0.42162022 0.59046058]
 [0.27753613 0.5155904  0.61417388]
 [0.33018912 0.55870007 0.58931918]
 [0.32793362 0.43080703 0.59555522]
 [0.34701218 0.46514272 0.60911494]
 [0.29693978 0.4925739  0.64452933]
```

```
 [0.32331457 0.48864241 0.51354555]]
Var: [0.08329162 0.08530521 0.08203152 0.08554274 0.07569542 0.06655688
 0.08464784 0.09350448]
P: [0.11181135 0.11537056 0.11811353 0.14177772 0.13988433 0.10624885
 0.12365976 0.14313389]
```

Test: output3

Output:

```
Input:
X: [[0.09542428 0.31836175]
 [0.88457197 0.05886196]
 [0.08824551 0.22137879]
 [0.69719126 0.56294657]
 [0.67936712 0.13337718]
 [0.77657018 0.15971943]
 [0.88326048 0.65115119]
 [0.93521449 0.064112  ]
 [0.55137788 0.77994441]
 [0.28454268 0.5385725 ]
 [0.91735044 0.10563432]]
K: 6
post:[[0.18336232 0.10941188 0.18054916 0.17033915 0.1564994  0.19983809]
 [0.2146522  0.11067785 0.24088393 0.13373411 0.05834415 0.24170776]
 [0.22263898 0.17803556 0.13117999 0.15664661 0.18169902 0.12979984]
 [0.27691789 0.27019538 0.17257473 0.11551992 0.08633457 0.07845751]
 [0.33264128 0.06985759 0.35468044 0.20387907 0.01770789 0.02123373]
 [0.19289655 0.12913821 0.24667229 0.24458903 0.01307949 0.17362443]
 [0.11575007 0.16509128 0.21063319 0.21313401 0.18907041 0.10632104]
 [0.17997691 0.16856807 0.11387806 0.18927847 0.173865   0.17443349]
 [0.13074034 0.13565889 0.18442253 0.14432638 0.21789258 0.18695928]
 [0.0140196  0.32175601 0.28739328 0.14190271 0.02292253 0.21200587]
 [0.38072876 0.36914003 0.02338031 0.08040019 0.04471169 0.10163902]]
Output:
Mu: [[0.66468099 0.26199962]
 [0.62569947 0.34485555]
 [0.60206122 0.34017529]
 [0.62235205 0.3213779 ]
 [0.5698483  0.40254431]
 [0.59161567 0.32746404]]
Var: [0.06810703 0.07662732 0.07048128 0.07606188 0.09451587 0.08469862]
P: [0.20402954 0.18432098 0.19511345 0.16306815 0.10564788 0.14782   ]
```

Test: output4

Output:

```
Input:
X: [[0.2768829  0.63528073 0.78995825]
 [0.09094997 0.26480029 0.88767477]
 [0.07477863 0.52775122 0.9243227 ]
 [0.20640906 0.07466311 0.2903126 ]
 [0.62908854 0.07187659 0.13985832]
 [0.71351669 0.97548082 0.5361715 ]
 [0.39626916 0.46020755 0.4044527 ]
 [0.63368997 0.8451724  0.91267846]
 [0.30188685 0.99086303 0.68130281]
 [0.25928876 0.13369046 0.15936088]
 [0.43420597 0.10155673 0.38943165]
 [0.96894215 0.91542484 0.73993596]]
K: 7
post:[[0.13319324 0.0843006  0.04206493 0.09418415 0.21353177 0.2423476
  0.19037771]
 [0.12447798 0.14484386 0.17675739 0.09884107 0.19989423 0.17427435
  0.08091113]
 [0.02839146 0.00244144 0.14717772 0.23456756 0.1770926  0.22723227
  0.18309696]
 [0.19119214 0.19070865 0.12662263 0.22587897 0.18695964 0.00093292
  0.07770506]
 [0.13160969 0.17301636 0.19595951 0.09874184 0.12411881 0.07868117
  0.19787262]
 [0.03584699 0.26199295 0.07917176 0.27367535 0.01735231 0.13423115
  0.1977295 ]
 [0.08500097 0.01560171 0.09499236 0.25306272 0.05412369 0.18126604
  0.3159525 ]
 [0.00988611 0.17600665 0.19733227 0.15515647 0.20604275 0.07254117
  0.18303458]
 [0.14675327 0.09056861 0.19155749 0.12814488 0.21526185 0.18134992
  0.04636398]
 [0.23493509 0.02948518 0.09860237 0.16683591 0.14947441 0.26372688
```

```
  0.05694014]
 [0.30489884 0.09641438 0.14286985 0.13031469 0.03349014 0.04072249
  0.25128961]
 [0.23982582 0.10673796 0.22547338 0.15137056 0.01043503 0.12712302
  0.13903424]]
Output:
Mu: [[0.42502212 0.39979602 0.48745511]
 [0.48817205 0.53389171 0.55923606]
 [0.44875493 0.51575663 0.59351583]
 [0.41795826 0.52189022 0.55982633]
 [0.31491102 0.48314983 0.62473251]
 [0.36371144 0.54329949 0.61147727]
 [0.45260565 0.4990613  0.56446244]]
Var: [0.08960266 0.09858954 0.09906158 0.08743994 0.08058401 0.08095448
 0.07952929]
P: [0.1388343  0.1143432  0.14321514 0.16756452 0.13231477 0.14370241
 0.16002567]
```

Test: output5

Output:

```
Input:
X: [[0.25217667 0.88883033 0.60945392]
 [0.1095267  0.17685611 0.74418757]
 [0.4638885  0.97958558 0.1913803 ]
 [0.90483937 0.13184594 0.42506922]
 [0.46394501 0.1469479  0.95700338]
 [0.72172065 0.16123525 0.25134604]
 [0.20332508 0.66775143 0.83018304]
 [0.83094798 0.41779628 0.68597613]
 [0.17238368 0.58320356 0.87834562]
 [0.57414079 0.16450605 0.66300048]
 [0.89673475 0.87578692 0.69157   ]
 [0.72850492 0.97507225 0.59019331]]
K: 8
post:[[0.04281147 0.14855736 0.07332457 0.25844222 0.17720888 0.03184671
  0.05142386 0.21638493]
 [0.12512158 0.26421715 0.05161975 0.24405635 0.03556194 0.11465112
  0.12513683 0.03963529]
 [0.20698621 0.1560979  0.03487289 0.05436257 0.04397632 0.14442513
  0.17955677 0.17972222]
 [0.10093276 0.16572063 0.04164508 0.14089246 0.21219396 0.13870304
  0.03974198 0.16017009]
 [0.16852121 0.16458453 0.12996509 0.13690618 0.18324299 0.11091361
  0.05199424 0.05387215]
 [0.08413964 0.05786269 0.07815442 0.16063314 0.21224124 0.13358694
  0.07869948 0.19468244]
 [0.06221241 0.16164303 0.10829022 0.13973763 0.20181911 0.10488237
  0.13720792 0.08420731]
 [0.06075997 0.16052971 0.20670337 0.248381   0.05933406 0.12684191
  0.13216981 0.00528016]
 [0.07118442 0.23855496 0.03504358 0.03334136 0.13569427 0.23231741
  0.14860732 0.10525668]
 [0.06146787 0.02764727 0.13654293 0.22390311 0.15630539 0.01249044
```

```
   0.23074302 0.15089997]
 [0.13864103 0.23677921 0.206487   0.00437927 0.09203464 0.13897102
  0.0657823  0.11692553]
 [0.09063853 0.19321564 0.25405123 0.0701173  0.1400906  0.18094611
  0.01129852 0.05964208]]
Output:
Mu: [[0.53717008 0.52985302 0.60170994]
 [0.49773159 0.55760274 0.66289779]
 [0.62147953 0.57181455 0.66134129]
 [0.4945142  0.41762258 0.63513601]
 [0.53781581 0.47136213 0.62992134]
 [0.53811797 0.54733784 0.62585717]
 [0.47102231 0.48659639 0.6272209 ]
 [0.53216583 0.54220432 0.54616378]]
Var: [0.0907923  0.08241078 0.06927684 0.07123489 0.07894573 0.08481847
 0.07540186 0.08508635]
P: [0.10111809 0.16461751 0.11305834 0.14292938 0.13747528 0.12254798
 0.1043635  0.1138899 ]
```

Hide output

Submit   You have used 1 of 20 attempts

✔ Correct (1/1 point)

Implementing run

1/1 point (graded)

Write a function `run` that runs the EM algorithm. The convergence criterion you should use is described above.

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

```
1 def run(X: np.ndarray, mixture: GaussianMixture,
2         post: np.ndarray) -> Tuple[GaussianMixture, np.ndarray, float]:
3     """Runs the mixture model
4
5     Args:
6         X: (n, d) array holding the data
7         post: (n, K) array holding the soft counts
8             for all components for all examples
9
10    Returns:
11        GaussianMixture: the new gaussian mixture
12        np.ndarray: (n, K) array holding the soft counts
13            for all components for all examples
14        float: log-likelihood of the current assignment
15    """
16    prev_log_likelihood = None
```

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 = -1036.8421
Cost = -1036.8418
```

Hide output

| Submit | You have used 1 of 20 attempts

✔ Correct (1/1 point)

Discussion

Hide Discussion

Topic: Unit 4 Unsupervised Learning (2 weeks) :Project 4: Collaborative Filtering via Gaussian Mixtures / 3. Expectation–maximization algorithm

Add a Post

Show all posts ▼	by recent activity ▼
💬 **e-step .. multivariate normal Scipy** Is anybody using Multivariate normal for this exercise? when trying to calculate the pdf for each value, since they have 2 dimensions, then I'm using …	2
❓ **How do you compute Multivariate Gaussian in Python?** How do you compute Multivariate Gaussian in Python? Is there a build-in function I can use?	2
💬 **error in estep** "ValueError: too many values to unpack (expected 2)" I received such error on estep. On stackoverflow I often see it was referred to iteration issues, …	3
☑ **[Staff] Could you please confirm that the log likelihood value provided to testing (-1388.0818, K = 3, seed = 0) is correct**	3

Dear Staff, I passed all functions tests in Part 3 and also all log likelihood values for EM Model in Part 4 were accepted by the grader. However my log…