GMM

June 4, 2021

[1]:
```python
import random
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Ellipse
import scipy as sc
from scipy import random, linalg, stats, special
```

[2]:
```python
#
NProperties = 1
NClasses = 4
NObjects = 200
distanceBTWclasses = 20
DiffBTWSpreadOFclasses = 2

#
Mu = [np.random.random(NProperties)*distanceBTWclasses*i for i in␣
 ↪range(1,NClasses+1)]
#
Var = [np.random.random(NProperties)*DiffBTWSpreadOFclasses*i for i in␣
 ↪range(1,NClasses+1)]
```

[5]:
```python
# (1)
#

theta = np.repeat(1.0/NClasses,NClasses)

print('    1 to '+str(NClasses))
print(theta)
```

```
    1 to 4
[0.25 0.25 0.25 0.25]
```

[6]:
```python
# (2)
#
r = np.random.multinomial(NObjects,theta)

print('     1 to '+str(NClasses))
print(r)
```

```
    1 to 4
[52 50 50 48]
```

[14]:

```
# (3)
#
rAlln = [np.random.normal(Mu[i], Var[i], r[i]) for i in range(0,NClasses)]

#     array
y = rAlln[0]
for i in range(NClasses-1):
    y = np.hstack((y,rAlln[i+1]))

#
v_true = np.zeros((1))
#     1 2 3 4
for i,j in enumerate(r):
    v_true = np.hstack((v_true, np.repeat(i+1, j)))
#   1
v_true = np.array(v_true[1:])
y_true = np.vstack((y, v_true))

#
np.random.shuffle(y_true.T)

# y
y = y_true[0,:]

print (y)
```

```
[35.46516911 59.97870443 36.50427966 56.86953831 31.87360191 10.81381695
  8.72865172 65.66798189 32.74487374 55.61182504 34.91328184  8.7920578
 32.62597648 34.05078789 34.91658404 34.06571831 40.50684612 33.99204709
 37.55039122 38.76424575 11.09563284 37.93596103 56.36610018 38.85005329
 36.90677749 34.17886747 58.05755236 65.10145135  8.23603301 33.0935772
 38.03240073 66.15893254  8.25844488 35.03642788 62.46199171 10.90899686
  8.18667972  8.09554166 32.43999732 60.22226079 36.18158612 56.22069
 11.16162734 57.85668468 39.20345304 34.50147923 34.22818315 34.88552932
 40.56929869 33.06961116 10.69515017 33.55083689 47.47347781 41.69704016
 53.19973111 53.97090252 33.32861132 40.5692706  37.92410818 53.74841492
 67.40205685 33.25395722  9.04207435 55.94382228 62.30866301 35.24007087
 59.10970427 35.90471043 40.28870931 10.59254467 13.27743509 60.35840441
 33.01246392 14.13380931 33.53121867 59.50813673 62.39880248 62.79855123
 33.40368541 31.7330905  33.20061598 10.31345449  8.66797075 55.82331895
 34.85427373 59.54131504 11.07463812 39.71460201  9.17108105 32.84705925
 33.67017874  7.12469152 63.38074998  8.96688662 40.01359752 39.55778699
 33.44797223 12.37174801 33.20506581 57.63679575  8.20393936 43.66868871
 10.22368964  7.0705856  34.85671121 36.5324913  11.18286768 40.39055692
 38.21794177 43.94295702 37.52921107 31.44577317 33.76263276 53.9947627
```

```
42.02069442 11.02595566 40.45956031 11.59341996 33.1264835  52.49981449
61.57463468 53.24237762 32.98196417 37.85467452 60.15600113 11.06052603
10.99795736  9.50695589 55.5557617  55.77040991 11.22468521  7.01359444
32.61646613 34.17001928 58.662715    9.80682219 39.19543209  7.13363496
11.06331833 33.13624218 37.93446805  7.53770841 60.66658641 31.54159915
 9.75133893 63.68908107 32.08185045 33.82248956 12.98256072 52.70843892
40.58127575 36.98851829 33.10618201  8.54580064 33.1224901   7.8205073
11.33825582 35.79930365 34.4846558  12.95377219 40.53693304 32.67811937
40.07778319 36.87867543 11.96071681 38.24773403 65.2425526  41.45880124
42.03290714 49.63986637  7.4088466  35.98437499 10.58205673 39.1129478
32.95553408 59.11004558 36.974044   61.56421716 45.56648576 12.2549322
 7.10510866 36.14057231 40.73866092 34.00807509 32.7443249   9.98012731
40.42455028 34.21280645 33.70772782 52.47133258 40.65464157 33.49753729
60.36892424 34.01337126  5.70199318 47.66712806 11.52996574 52.08130986
32.01032407 11.35736623]
```

[16]:
```
#  v
v = np.array([random.randint(1, NClasses+1) for i in range(y.shape[0])])
print(v)
```

```
[3 3 1 2 3 1 4 2 3 1 2 3 2 2 1 1 2 4 4 3 3 3 4 1 4 4 2 4 2 1 2 2 3 2 3 1 3
 3 2 2 3 1 1 1 1 1 2 1 4 4 4 1 1 3 3 2 1 3 1 2 2 4 2 2 3 1 3 1 3 4 4 2 2 3
 1 1 1 3 2 3 1 3 2 2 1 3 4 1 4 4 2 4 1 2 1 3 4 3 4 1 1 3 3 1 2 1 2 3 2 1 4
 3 2 2 4 4 4 2 3 3 2 2 1 2 3 1 2 3 3 1 2 2 1 4 2 3 2 2 3 2 1 1 4 2 1 2 2 2
 4 4 1 4 4 3 1 2 2 1 4 3 4 1 3 4 1 4 1 1 3 3 3 1 3 1 2 2 1 1 3 2 1 4 4 4 3
 1 4 4 3 4 4 4 4 1 3 3 3 2 2 3]
```

[21]:
```
## EM  ##
#
broadness = 15
initMu = np.random.random(NClasses)*max(y)
initVar = np.random.random(NClasses)+broadness
initW = theta #np.random.random(NClasses)
```

[22]:
```
# E-step
def EStep(y, w, Mu, Sigma):

    # r_ij
    r_ij = np.zeros((y.shape[0], Mu.shape[0]))

    for Object in range(y.shape[0]):

        r_ij_Sumj = np.zeros(Mu.shape[0])

        #    x
        for jClass in range(Mu.shape[0]):
            r_ij_Sumj[jClass] = w[jClass] * sc.stats.norm.pdf(y[Object],␣
 ↪Mu[jClass], np.sqrt(Sigma[jClass]))
```

```python
        #  x  j
        for jClass in range(r_ij_Sumj.shape[0]):
            r_ij[Object,jClass] =   r_ij_Sumj[jClass] / np.sum(r_ij_Sumj)

    return r_ij
```

[23]:
```python
r_n = EStep(y, initW, initMu, initVar)
print(r_n)
```

```
[[2.87733362e-18 8.03984545e-01 1.05926491e-13 1.96015455e-01]
 [1.62380842e-43 9.99999580e-01 2.10079918e-35 4.20105417e-07]
 [2.67324448e-19 8.76094136e-01 1.38381749e-14 1.23905864e-01]
 [2.74562491e-40 9.99997625e-01 1.24318805e-32 2.37530148e-06]
 [6.82573183e-15 3.87325588e-01 7.76396604e-11 6.12674412e-01]
 [7.14900182e-02 1.16009852e-09 9.28437664e-01 7.23165143e-05]
 [1.29711668e-01 1.58523820e-11 8.70285613e-01 2.71887133e-06]
 [1.96472588e-49 9.99999983e-01 1.75449816e-40 1.70065554e-08]
 [1.11779146e-15 4.98328018e-01 1.68971352e-11 5.01671982e-01]
 [5.53778120e-39 9.99995232e-01 1.64173512e-31 4.76761588e-06]
 [9.97983592e-18 7.54509165e-01 3.06620441e-13 2.45490835e-01]
 [1.27464441e-01 1.80733544e-11 8.72532553e-01 3.00613174e-06]
 [1.43512299e-15 4.82901837e-01 2.08678308e-11 5.17098163e-01]
 [6.75102929e-17 6.62110770e-01 1.56395394e-12 3.37889230e-01]
 [9.90628201e-18 7.54828606e-01 3.04689959e-13 2.45171394e-01]
 [6.53367207e-17 6.63849583e-01 1.52101265e-12 3.36150417e-01]
 [2.26092705e-23 9.83183906e-01 4.36913398e-18 1.68160940e-02]
 [7.67766621e-17 6.55228185e-01 1.74476472e-12 3.44771815e-01]
 [2.36324536e-20 9.24547069e-01 1.72494758e-15 7.54529308e-02]
 [1.37761562e-21 9.58744078e-01 1.49901805e-16 4.12559222e-02]
 [6.57755066e-02 2.06727644e-09 9.34112155e-01 1.12336476e-04]
 [9.60591415e-21 9.37547457e-01 7.95900162e-16 6.24525430e-02]
 [9.14036127e-40 9.99996860e-01 3.49301593e-32 3.14020790e-06]
 [1.12596844e-21 9.60500099e-01 1.26030002e-16 3.94999008e-02]
 [1.05467888e-19 8.97275835e-01 6.23091980e-15 1.02724165e-01]
 [5.09631772e-17 6.76882958e-01 1.23115062e-12 3.23117042e-01]
 [1.60564812e-41 9.99998773e-01 1.08530804e-33 1.22736286e-06]
 [7.64064279e-49 9.99999977e-01 5.62552455e-40 2.34549092e-08]
 [1.48330691e-01 5.71504057e-12 8.51668065e-01 1.24379905e-06]
 [5.34163010e-16 5.43509417e-01 9.04900063e-12 4.56490583e-01]
 [7.66598652e-21 9.40458052e-01 6.55635159e-16 5.95419477e-02]
 [6.05417608e-50 9.99999987e-01 6.39131675e-41 1.28664493e-08]
 [1.47438018e-01 5.98692305e-12 8.52560693e-01 1.28893810e-06]
 [7.57098714e-18 7.66232966e-01 2.42186344e-13 2.33767034e-01]
 [4.25979290e-46 9.99999896e-01 1.27851580e-37 1.04231247e-07]
 [6.95109020e-02 1.41012684e-09 9.30405176e-01 8.39209560e-05]
 [1.50312087e-01 5.15895756e-12 8.49686763e-01 1.14986581e-06]
 [1.54027871e-01 4.27003056e-12 8.45971135e-01 9.94564181e-07]
```

```
[2.11747915e-15 4.58858790e-01 2.89756895e-11 5.41141210e-01]
[9.06761319e-44 9.99999633e-01 1.27390249e-35 3.66572976e-07]
[5.61470363e-19 8.56518204e-01 2.61442559e-14 1.43481796e-01]
[1.29362511e-39 9.99996596e-01 4.70738140e-32 3.40366115e-06]
[6.44995600e-02 2.36659642e-09 9.35375907e-01 1.24530987e-04]
[2.59515639e-41 9.99998627e-01 1.63918056e-33 1.37252131e-06]
[4.90189853e-22 9.67008204e-01 6.16391004e-17 3.29917962e-02]
[2.49920900e-17 7.12543903e-01 6.70989213e-13 2.87456097e-01]
[4.57218326e-17 6.82481407e-01 1.12250203e-12 3.17518593e-01]
[1.06197482e-17 7.51813544e-01 3.23328574e-13 2.48186456e-01]
[1.95048800e-23 9.83723281e-01 3.84770604e-18 1.62767186e-02]
[5.62119193e-16 5.40417072e-01 9.44831057e-12 4.59582928e-01]
[7.40297873e-02 9.09453088e-10 9.25910148e-01 6.00637688e-05]
[2.00347074e-16 6.01643462e-01 3.94113101e-12 3.98356538e-01]
[1.47830902e-30 9.99591151e-01 2.87557468e-24 4.08849254e-04]
[1.35049785e-24 9.90997332e-01 3.86624460e-19 9.00266820e-03]
[1.75268979e-36 9.99981980e-01 2.31075728e-29 1.80204635e-05]
[2.78338109e-37 9.99988209e-01 4.75352215e-30 1.17910069e-05]
[3.23234273e-16 5.73633368e-01 5.91294704e-12 4.26366632e-01]
[1.95061760e-23 9.83723043e-01 3.84792602e-18 1.62769574e-02]
[9.87586166e-21 9.37180755e-01 8.15083480e-16 6.28192453e-02]
[4.73317372e-37 9.99986673e-01 7.50166491e-30 1.33271499e-05]
[3.06969294e-51 9.99999994e-01 4.95306424e-42 6.33851544e-09]
[3.79301498e-16 5.64111079e-01 6.77132919e-12 4.35888921e-01]
[1.18923516e-01 3.02969264e-11 8.81072019e-01 4.46479329e-06]
[2.50611118e-39 9.99996032e-01 8.30788109e-32 3.96759590e-06]
[6.14940396e-46 9.99999886e-01 1.75207105e-37 1.13629198e-07]
[4.78684635e-18 7.84760914e-01 1.63690786e-13 2.15239086e-01]
[1.29771545e-42 9.99999317e-01 1.25143946e-34 6.82753821e-07]
[1.05831820e-18 8.37744615e-01 4.50008332e-14 1.62255385e-01]
[3.78662967e-23 9.81159395e-01 6.80937411e-18 1.88406047e-02]
[7.62919886e-02 7.36787083e-10 9.23656859e-01 5.11517904e-05]
[3.38912394e-02 1.78288648e-07 9.62774733e-01 3.33384929e-03]
[6.54691222e-44 9.99999660e-01 9.63146716e-36 3.39666998e-07]
[6.34729132e-16 5.33032068e-01 1.04714000e-11 4.66967932e-01]
[2.57758350e-02 1.01298992e-06 9.61785316e-01 1.24378357e-02]
[2.09015595e-16 5.99193795e-01 4.08536458e-12 4.00806205e-01]
[5.00448966e-43 9.99999453e-01 5.52186172e-35 5.46543700e-07]
[4.95563510e-46 9.99999892e-01 1.45580438e-37 1.08006730e-07]
[1.90270134e-46 9.99999914e-01 6.40189773e-38 8.62282169e-08]
[2.75104157e-16 5.83156487e-01 5.15752834e-12 4.16843513e-01]
[9.09621096e-15 3.70204753e-01 9.88298844e-11 6.29795247e-01]
[4.25128938e-16 5.57279027e-01 7.45828268e-12 4.42720973e-01]
[8.27671507e-02 4.15403878e-10 9.17199816e-01 3.30326462e-05]
[1.31893608e-01 1.39823496e-11 8.68103922e-01 2.46958549e-06]
[3.34185236e-39 9.99995759e-01 1.06380690e-31 4.24121903e-06]
[1.13891504e-17 7.48754273e-01 3.43222076e-13 2.51245727e-01]
[4.62271597e-43 9.99999463e-01 5.15816595e-35 5.36503982e-07]
```

```
[6.61862765e-02 1.98022217e-09 9.33705009e-01 1.08712507e-04]
[1.46914477e-22 9.74628894e-01 2.18627034e-17 2.53711061e-02]
[1.14711977e-01 3.95421282e-11 8.85282548e-01 5.47423265e-06]
[9.01062920e-16 5.11593762e-01 1.40832625e-11 4.88406238e-01]
[1.54754668e-16 6.16434748e-01 3.16540605e-12 3.83565252e-01]
[1.98331414e-01 5.65665005e-13 8.01668376e-01 2.10412251e-07]
[4.71824743e-47 9.99999938e-01 1.93466155e-38 6.20907746e-08]
[1.21438944e-01 2.59391498e-11 8.78557091e-01 3.96434187e-06]
[7.25328349e-23 9.78264892e-01 1.19119789e-17 2.17351085e-02]
[2.12672741e-22 9.72492910e-01 3.00541950e-17 2.75070901e-02]
[2.50101139e-16 5.88746711e-01 4.75716054e-12 4.11253289e-01]
[4.48103232e-02 2.81252369e-08 9.54370035e-01 8.19613501e-04]
[4.21106000e-16 5.57849797e-01 7.39845753e-12 4.42150203e-01]
[4.38937683e-41 9.99998449e-01 2.57423159e-33 1.55107447e-06]
[1.49616711e-01 5.34700843e-12 8.50382107e-01 1.18188490e-06]
[1.25533722e-26 9.96839813e-01 6.90071373e-21 3.16018732e-03]
[8.49531864e-02 3.45441452e-10 9.15018119e-01 2.86944731e-05]
[2.01060234e-01 5.05198968e-13 7.98939573e-01 1.92879889e-07]
[1.13272152e-17 7.48993740e-01 3.41628232e-13 2.51006260e-01]
[2.50492122e-19 8.77692558e-01 1.30874758e-14 1.22307442e-01]
[6.40937808e-02 2.47185342e-09 9.35777487e-01 1.28729896e-04]
[2.97644939e-23 9.82132630e-01 5.53534118e-18 1.78673699e-02]
[4.96475034e-21 9.45705498e-01 4.51332851e-16 5.42945019e-02]
[6.54394190e-27 9.97270381e-01 3.93953235e-21 2.72961942e-03]
[2.48284579e-20 9.23765665e-01 1.79967248e-15 7.62343346e-02]
[1.63009647e-14 3.36250417e-01 1.61267745e-10 6.63749583e-01]
[1.26617923e-16 6.27752133e-01 2.66939062e-12 3.72247867e-01]
[2.62932003e-37 9.99988363e-01 4.52653153e-30 1.16371104e-05]
[6.27027988e-25 9.92412325e-01 1.99776151e-19 7.58767478e-03]
[6.71478793e-02 1.79218855e-09 9.32751367e-01 1.00751480e-04]
[2.52840236e-23 9.82763934e-01 4.81037769e-18 1.72360663e-02]
[5.67000111e-02 5.72916718e-09 9.43055765e-01 2.44218513e-04]
[4.97997699e-16 5.47750162e-01 8.52754672e-12 4.52249838e-01]
[9.30648739e-36 9.99973538e-01 9.70337163e-29 2.64623694e-05]
[3.56455569e-45 9.99999828e-01 7.91767108e-37 1.71698364e-07]
[1.58313779e-36 9.99982397e-01 2.11729102e-29 1.76030806e-05]
[6.77185333e-16 5.29085006e-01 1.10610427e-11 4.70914994e-01]
[1.16161448e-20 9.34992238e-01 9.37072668e-16 6.50077621e-02]
[1.06251625e-43 9.99999620e-01 1.45962084e-35 3.80424972e-07]
[6.64637122e-02 1.92377365e-09 9.33429944e-01 1.06342315e-04]
[6.77067276e-02 1.69223545e-09 9.32196832e-01 9.64390248e-05]
[1.04348742e-01 7.90441002e-11 8.95641960e-01 9.29872721e-06]
[6.33108531e-39 9.99995082e-01 1.84184732e-31 4.91773857e-06]
[3.79195356e-39 9.99995633e-01 1.18578379e-31 4.36721503e-06]
[6.33017738e-02 2.69293806e-09 9.36560810e-01 1.37413552e-04]
[2.03964629e-01 4.48462677e-13 7.96035195e-01 1.75980607e-07]
[1.46403742e-15 4.81669169e-01 2.12222560e-11 5.18330831e-01]
[5.19644530e-17 6.75873124e-01 1.25170960e-12 3.24126876e-01]
```

```
[3.77879511e-42 9.99999124e-01 3.13324867e-34 8.76112065e-07]
[9.58005909e-02 1.46576001e-10 9.04184501e-01 1.49082705e-05]
[4.99537016e-22 9.66872615e-01 6.26487214e-17 3.31273850e-02]
[1.97883028e-01 5.76332238e-13 8.02116758e-01 2.13459012e-07]
[6.64087322e-02 1.93481391e-09 9.33484459e-01 1.06807171e-04]
[4.87744138e-16 5.49006569e-01 8.37864502e-12 4.50993431e-01]
[9.63950738e-21 9.37501377e-01 7.98291504e-16 6.24986230e-02]
[1.78412827e-01 1.33880142e-12 8.21586765e-01 4.08140859e-07]
[3.13184212e-44 9.99999714e-01 5.11411478e-36 2.85802800e-07]
[1.34263599e-14 3.47405217e-01 1.37044899e-10 6.52594783e-01]
[9.73338265e-02 1.30756938e-10 9.02652511e-01 1.36623452e-05]
[2.25426596e-47 9.99999948e-01 1.02645573e-38 5.21682943e-08]
[4.44948958e-15 4.13221656e-01 5.41694638e-11 5.86778344e-01]
[1.11159561e-16 6.35007699e-01 2.38982502e-12 3.64992301e-01]
[3.71462857e-02 9.77912110e-08 9.60740535e-01 2.11308126e-03]
[5.65824766e-36 9.99976399e-01 6.32683219e-29 2.36006992e-05]
[1.89601130e-23 9.83824771e-01 3.75505225e-18 1.61752291e-02]
[8.72681024e-20 9.01167964e-01 5.29598180e-15 9.88320359e-02]
[5.20013930e-16 5.45134574e-01 8.84566725e-12 4.54865426e-01]
[1.36380412e-01 1.08585618e-11 8.63617554e-01 2.03459713e-06]
[5.02254578e-16 5.47235856e-01 8.58922408e-12 4.52764144e-01]
[1.65694356e-01 2.41158405e-12 8.34305003e-01 6.41448334e-07]
[6.11962177e-02 3.39816417e-09 9.38639721e-01 1.64057850e-04]
[1.34614800e-18 8.30098011e-01 5.52947712e-14 1.69901989e-01]
[2.59420460e-17 7.10742272e-01 6.92668294e-13 2.89257728e-01]
[3.74786533e-02 9.22184358e-08 9.60500276e-01 2.02097860e-03]
[2.10565050e-23 9.83445900e-01 4.10966694e-18 1.65540997e-02]
[1.28630700e-15 4.89664136e-01 1.90251599e-11 5.10335864e-01]
[6.23297607e-23 9.78976648e-01 1.04553113e-17 2.10233521e-02]
[1.12560227e-19 8.95906907e-01 6.58883820e-15 1.04093093e-01]
[5.07645518e-02 1.21438782e-08 9.48802776e-01 4.32659619e-04]
[4.63000057e-21 9.46506630e-01 4.25047172e-16 5.34933697e-02]
[5.44801773e-49 9.99999978e-01 4.20864087e-40 2.16510816e-08]
[2.37500591e-24 9.89792483e-01 6.28453151e-19 1.02075169e-02]
[6.09131268e-25 9.92461183e-01 1.94859308e-19 7.53881662e-03]
[8.50025408e-33 9.99873771e-01 3.40481275e-26 1.26229320e-04]
[1.84455063e-01 1.02351551e-12 8.15544605e-01 3.32018695e-07]
[8.82130165e-19 8.43339254e-01 3.85021449e-14 1.56660746e-01]
[7.65267382e-02 7.21097235e-10 9.23422942e-01 5.03186525e-05]
[6.06612890e-22 9.65446881e-01 7.40384069e-17 3.45531191e-02]
[7.16230126e-16 5.25661926e-01 1.15982090e-11 4.74338074e-01]
[1.29665671e-42 9.99999317e-01 1.25056272e-34 6.82623757e-07]
[9.02460151e-20 9.00488399e-01 5.45072617e-15 9.95116007e-02]
[3.65455867e-45 9.99999827e-01 8.08896870e-37 1.72706230e-07]
[1.38050100e-28 9.98856722e-01 1.42423566e-22 1.14327768e-03]
[4.64328337e-02 2.21553211e-08 9.52883538e-01 6.83606187e-04]
[1.99315863e-01 5.42989787e-13 8.00683933e-01 2.03891014e-07]
[6.16852017e-19 8.53855071e-01 2.83391285e-14 1.46144929e-01]
```

```
[1.30662351e-23 9.85102221e-01 2.72573263e-18 1.48977792e-02]
[7.41304508e-17 6.57112668e-01 1.69348594e-12 3.42887332e-01]
[1.11908407e-15 4.98256767e-01 1.69136428e-11 5.01743233e-01]
[9.11489310e-02 2.09370656e-10 9.08831492e-01 1.95764538e-05]
[2.74660238e-23 9.82446404e-01 5.16549903e-18 1.75535964e-02]
[4.72962250e-17 6.80741278e-01 1.15532608e-12 3.19258722e-01]
[1.42652092e-16 6.21046813e-01 2.95390641e-12 3.78953187e-01]
[9.96064366e-36 9.99973121e-01 1.02867708e-28 2.68789464e-05]
[1.59395179e-23 9.84433070e-01 3.23418647e-18 1.55669297e-02]
[2.24768612e-16 5.94976904e-01 4.34515514e-12 4.05023096e-01]
[6.38419053e-44 9.99999662e-01 9.42558470e-36 3.37671704e-07]
[7.32760070e-17 6.57734296e-01 1.67686947e-12 3.42265704e-01]
[2.79247388e-01 2.84787567e-14 7.20752591e-01 2.09833502e-08]
[9.32374877e-31 9.99631813e-01 1.93444163e-24 3.68186756e-04]
[5.77885547e-02 5.03145645e-09 9.41990218e-01 2.21222721e-04]
[2.52487409e-35 9.99966715e-01 2.28814399e-28 3.32851906e-05]
[5.15557131e-15 4.04263405e-01 6.13175163e-11 5.95736595e-01]
[6.08483555e-02 3.53377928e-09 9.38982619e-01 1.69022417e-04]]
```

[24]:
```python
# M-step
def MStep(r, y, Mu, Sigma):

    N = y.shape[0]

    mu_j = np.zeros((N, Mu.shape[0]))
    sigma_j = np.zeros((N, Mu.shape[0]))

    for Object in range(y.shape[0]):

        #
        mu_j[Object,:] = r[Object,:] * y[Object]

        #
        sigma_j[Object,:] = r[Object,:] * np.square(-Mu + y[Object])

    w_j = np.sum(r, axis=0) / N
    mu_j = (1/np.sum(r, axis=0)) * np.sum(mu_j, axis=0)
    sigma_j = (1/np.sum(r, axis=0)) * np.sum(sigma_j, axis=0)

    return w_j,mu_j,sigma_j
```

[26]:
```python
w_n,mu_n,sigma_n = MStep(r_n, y, initMu, initVar)

print(w_n)
print(mu_n)
print(sigma_n)
```

```
[0.02733617 0.61652775 0.23254317 0.12359292]
```

```
[ 8.93582158 45.02592879 10.02934141 33.62986437]
[ 78.63121389 189.18362086  27.75852241  26.89792195]
```

[29]:
```
#
Inititeration = 10
# EM
EMiteration = 200
lookLH = 20

for init in range(Inititeration):

    #
    initMu = np.random.random(NClasses)*max(y)
    r_n = EStep(y, initW, initMu, initVar)
    w_n,mu_n,sigma_n = MStep(r_n, y, initMu, initVar)

    if init == 0:
        logLH = -1000000000000

    for i in range(EMiteration):

        # E-step
        r_n = EStep(y, w_n, mu_n, sigma_n)

        # M-step
        w_n,mu_n,sigma_n = MStep(r_n, y, mu_n, sigma_n)

        #
        logLall = np.zeros((y.shape[0]))

        for Object in range(y.shape[0]):

            LH = np.zeros(NClasses)

            for jClass in range(NClasses):
                LH[jClass] = w_n[jClass] * sc.stats.norm.pdf(y[Object],␣
↪mu_n[jClass], np.sqrt(sigma_n[jClass]))

            logLall[Object] = np.log(np.sum(LH))

        logL = np.sum(logLall)

        if i > EMiteration - lookLH:
            print(logL)

    #
    if logL > logLH:
```

```
        logLH = logL
        print('found larger: ', logLH)
        w_p = w_n
        mu_p = mu_n
        sigma_p = sigma_n
        r_p = r_n
```

```
-717.2906434129388
-717.2906430186656
-717.290642623466
-717.2906422273386
-717.2906418302824
-717.2906414322961
-717.2906410333783
-717.2906406335276
-717.2906402327428
-717.2906398310224
-717.2906394283648
-717.2906390247687
-717.2906386202322
-717.290638214754
-717.2906378083323
-717.2906374009655
-717.2906369926516
-717.2906365833888
-717.2906361731755
found larger:  -717.2906361731755
-716.6837752128406
-716.5430168040427
-716.3752628378284
-716.1851340291353
-715.9838450522903
-715.7872914204823
-715.6099554125049
-715.4584584443882
-715.3304473973323
-715.2191833217892
-715.1187693590323
-715.0263965056679
-714.9419584915969
-714.866696878275
-714.8019198290726
-714.7481975849782
-714.7051234363582
-714.6715150801884
-714.6458030053689
found larger:  -714.6458030053689
-696.7281976251924
```

```
-696.7281976251924
-696.7281976251923
-696.7281976251924
-696.7281976251925
-696.7281976251925
-696.7281976251924
-696.7281976251924
-696.7281976251925
-696.7281976251924
-696.7281976251924
-696.7281976251925
-696.7281976251924
-696.7281976251925
-696.7281976251925
-696.7281976251925
-696.7281976251925
-696.7281976251925
-696.7281976251925
found larger:  -696.7281976251925
-716.5865501537912
-716.5865369642561
-716.5865240933423
-716.586511534257
-716.5864992803156
-716.5864873249424
-716.5864756616694
-716.586464284137
-716.5864531860929
-716.586442361391
-716.5864318039921
-716.5864215079623
-716.5864114674723
-716.5864016767977
-716.5863921303168
-716.5863828225104
-716.586373747961
-716.5863649013523
-716.5863562774673
-696.7281976251923
-696.7281976251925
-696.7281976251924
-696.7281976251924
-696.7281976251925
-696.7281976251925
-696.7281976251925
-696.7281976251925
-696.7281976251925
-696.7281976251924
```

-696.7281976251925
-696.7281976251925
-696.7281976251924
-696.7281976251924
-696.7281976251924
-696.7281976251925
-696.7281976251924
-696.7281976251925
-696.7281976251925
-696.7281989985008
-696.7281986611692
-696.7281984066991
-696.728198214736
-696.7281980699258
-696.7281979606859
-696.7281978782787
-696.7281978161134
-696.7281977692177
-696.7281977338411
-696.728197707154
-696.728197687022
-696.7281976718349
-696.7281976603782
-696.7281976517356
-696.7281976452159
-696.7281976402976
-696.7281976365873
-696.7281976337885
-717.2907312121165
-717.2907311691579
-717.2907311257445
-717.2907310818716
-717.2907310375344
-717.2907309927283
-717.2907309474482
-717.2907309016894
-717.290730855447
-717.2907308087158
-717.2907307614912
-717.290730713768
-717.2907306655409
-717.2907306168048
-717.2907305675546
-717.290730517785
-717.2907304674907
-717.2907304166665
-717.2907303653066
-711.5259057167126

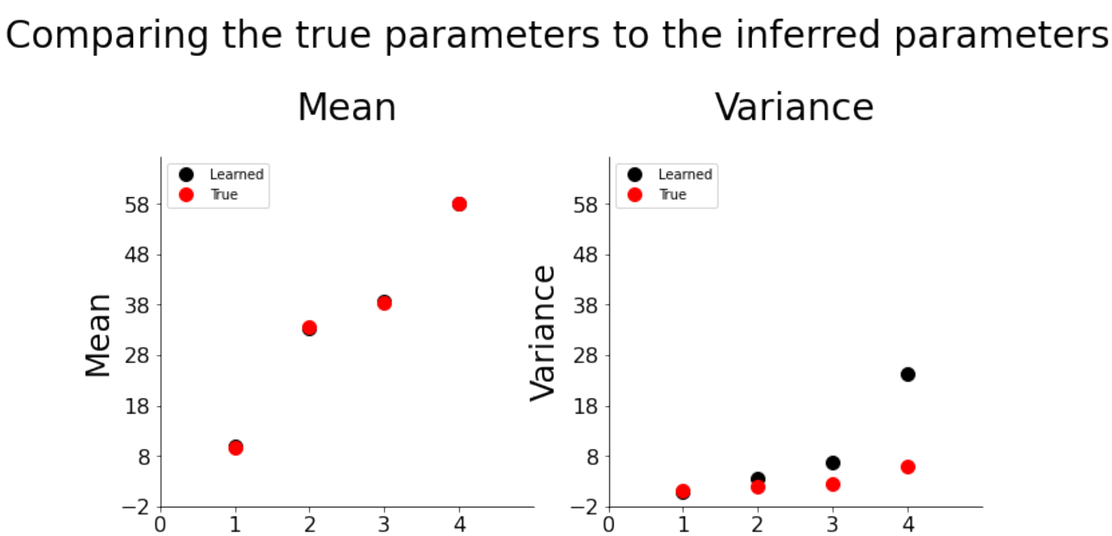
-711.5257178136394
-711.5256008099693
-711.5255286093775
-711.5254843681321
-711.5254574068475
-711.5254410457576
-711.5254311498361
-711.525425179563
-711.5254215847547
-711.5254194235583
-711.5254181257824
-711.5254173471938
-711.5254168804169
-711.5254166007294
-711.5254164332147
-711.5254163329171
-711.5254162728801
-711.5254162369499
-781.6277826015228
-781.6277457707688
-781.627708908015
-781.627672010884
-781.6276350769754
-781.6275981038655
-781.6275610891067
-781.6275240302266
-781.627486924727
-781.6274497700838
-781.6274125637457
-781.6273753031335
-781.6273379856395
-781.6273006086271
-781.6272631694292
-781.6272256653479
-781.6271880936536
-781.6271504515844
-781.6271127363445
-716.6753446015512
-716.4090031422725
-716.0999553707853
-715.8312449779576
-715.6532837016493
-715.5465280624604
-715.4741592607286
-715.4135829179539
-715.355947831134
-715.298920880249
-715.2421919840776

```
-715.1859407600878
-715.1305134800082
-715.0763409350674
-715.0238840247771
-714.9735954090733
-714.9258937692869
-714.8811446443922
-714.8396449841845
```

[33]:

```
#
Mu_inf = np.sort(mu_p)
Mu_true = np.sort([Mu[i][0] for i in range(len(Mu))])

Var_inf = np.sort(sigma_p)
Var_true = np.sort([Var[i][0] for i in range(len(Var))])


plotsize = 11
sizeMean = 20
text_size = 16
axis_font = {'fontname':'Arial', 'size':'24'}
Title_font = {'fontname':'Arial', 'size':'28'}
x = range(1,NClasses+1)
startx = 0
endx = 5
stepsizex = 1
starty = -2
endy = max(y)
stepsizey = 10


fig = plt.figure()
ax1 = fig.add_subplot(2,2,1)
ax2 = fig.add_subplot(2,2,2)

ax1.plot(x, Mu_inf, 'k.', markersize=sizeMean, label='Learned')
ax1.plot(x, Mu_true, 'r.', markersize=sizeMean, label='True')

ax2.plot(x, Var_inf, 'k.', markersize=sizeMean, label='Learned')
ax2.plot(x, Var_true, 'r.', markersize=sizeMean, label='True')

for label in (ax1.get_xticklabels() + ax1.get_yticklabels()):
    label.set_fontname('Arial')
    label.set_fontsize(text_size)
ax1.spines['right'].set_visible(False)
ax1.spines['top'].set_visible(False)
ax1.xaxis.set_ticks_position('bottom')
```

```
ax1.yaxis.set_ticks_position('left')
ax1.xaxis.set_ticks(np.arange(startx, endx, stepsizex))
ax1.yaxis.set_ticks(np.arange(starty, endy, stepsizey))
ax1.set_xlim([startx, endx])
ax1.set_ylim([starty, endy])
ax1.set_ylabel('Mean', **axis_font)
ax1.legend(loc='upper left',fontsize=text_size-6)
ax1.set_title('Mean', y=1.08, **Title_font)
ax1.figure.set_size_inches(plotsize,plotsize)

for label in (ax2.get_xticklabels() + ax2.get_yticklabels()):
    label.set_fontname('Arial')
    label.set_fontsize(text_size)
ax2.spines['right'].set_visible(False)
ax2.spines['top'].set_visible(False)
ax2.xaxis.set_ticks_position('bottom')
ax2.yaxis.set_ticks_position('left')
ax2.xaxis.set_ticks(np.arange(startx, endx, stepsizex))
ax2.yaxis.set_ticks(np.arange(starty, endy, stepsizey))
ax2.set_xlim([startx, endx])
ax2.set_ylim([starty, endy])
ax2.set_ylabel('Variance', **axis_font)
ax2.legend(loc='upper left',fontsize=text_size-6)
ax2.set_title('Variance', y=1.08, **Title_font)
ax2.figure.set_size_inches(plotsize,plotsize)

plt.suptitle('Comparing the true parameters to the inferred␣
↪parameters',**Title_font)
fig.subplots_adjust(top=0.85)
plt.show()
```



```
[ ]:
```