Q1

June 17, 2020

```
In [41]: import numpy as np

In [16]: # part 1
         lmbda = np.array([39.288, 10.676])

         u1 = np.array([[0.5719, 0.1469],
                        [0.5885, 0.9817],
                        [0.5715, -0.1210]])

         u2 = np.array([[0.5121, -0.4042],
                        [0.6284, 0.5877],
                        [0.5856, 0.7009]])

         u3 = np.array([[0.5605, -0.3179 ],
                        [0.4921, -0.3682 ],
                        [0.6661, 0.8737]])

         u4 = np.array([[0.7502, -0.9201],
                        [0.6612, 0.3917]])

In [36]: u11 = u1[:,0]
         u21 = u2[:,0]
         u31 = u3[:,0]
         u41 = u4[:,0]

         u12 = u1[:,1]
         u22 = u2[:,1]
         u32 = u3[:,1]
         u42 = u4[:,1]

In [34]: # part a
         np.einsum('i,j',u11,u21)

Out[34]: array([[0.29286999, 0.35938196, 0.33490464],
               [0.30137085, 0.3698134 , 0.3446256 ],
               [0.29266515, 0.3591306 , 0.3346704 ]])
```

$$U_{1,1} \circ U_{2,1} = \begin{bmatrix} .5719x.5121 & .5719x.6284 & .5719x.5856 \\ .5885x.5121 & .5885x.6284 & .5885x.5856 \\ .5715x.5121 & .5715x.6284 & .5715x.5856 \end{bmatrix} = \begin{bmatrix} 0.29286999 & 0.35938196 & 0.33490464 \\ 0.30137085 & 0.3698134 & 0.3446256 \\ 0.29266515 & 0.3591306 & 0.3346704 \end{bmatrix}$$

```
In [37]: #part b
         b1=lmbda[0] * np.einsum('i,j,k,l',u11,u21,u31, u41)
         b2=lmbda[1] * np.einsum('i,j,k,l',u12,u22,u32, u42)
         b3= b1+b2
```

```
In [38]: b1
```

```
Out[38]: array([[[[4.8382407 , 4.26425586],
                  [4.24781132, 3.7438721 ],
                  [5.74978078, 5.06765536]],

                 [[5.93702491, 5.23268577],
                  [5.21250661, 4.59412073],
                  [7.05557946, 6.21854058]],

                 [[5.5326572 , 4.87629024],
                  [4.85748547, 4.28121754],
                  [6.57502758, 5.79499899]]],


                [[[4.97867573, 4.38803038],
                  [4.37110852, 3.85254193],
                  [5.91667423, 5.2147494 ]],

                 [[6.10935331, 5.38456999],
                  [5.36380511, 4.72746992],
                  [7.26037509, 6.39904027]],

                 [[5.6932484 , 5.01782971],
                  [4.99847911, 4.40548439],
                  [6.76587469, 5.96320494]]],


                [[[4.83485672, 4.26127335],
                  [4.24484031, 3.74125355],
                  [5.74575925, 5.06411093]],

                 [[5.93287241, 5.22902591],
                  [5.20886087, 4.5909075 ],
                  [7.05064463, 6.21419119]],

                 [[5.52878753, 4.87287965],
                  [4.85408804, 4.27822315],
                  [6.57042886, 5.79094583]]]])
```

```
In [39]: b2
```

```
Out[39]: array([[[[-0.18541814,  0.07893521],
                  [-0.21475609,  0.0914248 ],
                  [ 0.50959368, -0.21694147]],

                 [[ 0.26959486, -0.11477047],
                  [ 0.31225174, -0.13293012],
                  [-0.74094064,  0.31542925]],

                 [[ 0.32152295, -0.13687701],
                  [ 0.3723962 , -0.1585345 ],
                  [-0.88365713,  0.37618574]]],


                [[[-1.23910818,  0.52750644],
                  [-1.43516713,  0.6109716 ],
                  [ 3.40550115, -1.44977155]],

                 [[ 1.80164245, -0.76698549],
                  [ 2.08670887, -0.88834242],
                  [-4.95154139,  2.10794344]],

                 [[ 2.14866631, -0.91471861],
                  [ 2.48864088, -1.05945075],
                  [-5.90528391,  2.51396556]]],


                [[[ 0.15272699, -0.06501811],
                  [ 0.17689235, -0.07530566],
                  [-0.41974701,  0.17869243]],

                 [[-0.22206248,  0.09453524],
                  [-0.25719851,  0.10949316],
                  [ 0.61030509, -0.25981579]],

                 [[-0.26483511,  0.11274417],
                  [-0.30673887,  0.13058321],
                  [ 0.72785918, -0.30986028]]]])
```

```
In [40]: b3
```

```
Out[40]: array([[[[4.65282255, 4.34319107],
                  [4.03305524, 3.8352969 ],
                  [6.25937447, 4.85071389]],

                 [[6.20661977, 5.11791531],
                  [5.52475835, 4.46119061],
                  [6.31463882, 6.53396982]],

                 [[5.85418015, 4.73941323],
                  [5.22988167, 4.12268304],
                  [5.69137045, 6.17118473]]],


                [[[3.73956755, 4.91553682],
                  [2.93594139, 4.46351353],
                  [9.32217538, 3.76497785]],

                 [[7.91099576, 4.6175845 ],
                  [7.45051398, 3.8391275 ],
                  [2.30883371, 8.50698371]],

                 [[7.84191472, 4.10311109],
                  [7.48711999, 3.34603364],
                  [0.86059077, 8.47717049]]],


                [[[4.98758371, 4.19625523],
                  [4.42173266, 3.66594789],
                  [5.32601224, 5.24280336]],

                 [[5.71080993, 5.32356115],
                  [4.95166236, 4.70040065],
                  [7.66094972, 5.9543754 ]],

                 [[5.26395243, 4.98562382],
                  [4.54734917, 4.40880637],
                  [7.29828804, 5.48108555]]]])
```

```
In [69]: # part 2
         g11 = np.array([[38.946, 0.8653],
                         [0.9666, -4.8832]])

         g21 = np.array([[-0.4799, -0.0792],
                         [-1.7302, -4.3675]])

         g12 = np.array([[0.7059, -1.6496],
                         [0.7553, -1.1648]])

         g22 = np.array([[5.7493, -3.3204],
                         [-2.0019, 7.6587]])

         u1 = np.array([[0.5661, -0.1945],
                        [0.6005, -0.5685],
                        [0.5648, 0.7994]])
```

```
        u2 = np.array([[0.5031, 0.8331],
                       [0.6345, -0.1755],
                       [0.5867, -0.5246]])

        u3 = np.array([[0.5773, -0.3364],
                       [0.5013, -0.5733],
                       [0.6445, 0.7471]])

        u4 = np.array([[0.7524, -0.658 ],
                       [0.6587, 0.7524]])
```

```
In [133]: c1 = np.array([g11,g21])
          c2 = np.array([g12,g22])
          g = np.array([c1,c2])
          g.shape

Out[133]: (2, 2, 2, 2)

In [134]: tucker = tensorly.tucker_to_tensor(g,[u1,u2,u3,u4])

In [136]: tucker.shape

Out[136]: (3, 3, 3, 2)

In [135]: tucker

Out[135]: array([[[[ 3.00005471,  5.3689007 ],
                   [ 1.81956239,  5.68430554],
                   [ 6.48561691,  1.91262153]],

                  [[ 5.69605257,  5.70916416],
                   [ 4.701262  ,  5.08490697],
                   [ 7.33696029,  5.86532227]],

                  [[ 5.83203974,  4.96526222],
                   [ 5.05816843,  4.08603546],
                   [ 6.53526251,  6.4438178 ]]],


                 [[[ 1.2956372 ,  5.98013323],
                   [-0.19517582,  6.63479675],
                   [ 6.71764191,  0.91921501]],

                  [[ 6.30977072,  6.08204755],
                   [ 5.4225399 ,  5.31253386],
                   [ 7.27009883,  6.66556679]],

                  [[ 7.21603762,  5.19245382],
                   [ 6.6890497 ,  4.00956391],
                   [ 6.36721486,  7.79044013]]],


                 [[[ 8.16839446,  4.57487163],
                   [ 7.64504902,  4.01154602],
                   [ 6.91531628,  4.95191776]],

                  [[ 4.94898436,  5.62485519],
                   [ 3.49563136,  5.29641449],
                   [ 8.72644606,  4.6344422 ]],

                  [[ 2.99446728,  5.15829595],
                   [ 1.41618797,  4.96747811],
                   [ 8.07051237,  3.8093327 ]]]])
```

```
In [94]: # part 3
         x11 = np.array([[4, 0, 9],
                         [7, 9, 9],
                         [4, 8, 5]])

         x21 = np.array([[7, 8, 2],
                         [1, 5, 8],
                         [7, 9, 2]])

         x31 = np.array([[7, 9, 4],
                         [10, 1, 2],
                         [1, 5, 8]])

         x12 = np.array([[6, 5, 1],
                         [3, 3, 5],
                         [1, 8, 7]])

         x22 = np.array([[8, 2, 3],
                         [4, 3, 3],
                         [2, 4, 6]])

         x23 = np.array([[6, 6, 8],
                         [5, 9, 8],
                         [3, 9, 5]])
```

```
In [156]: c1 = np.array([x11,x21,x31])
          c2 = np.array([x12,x22,x23])
          X = np.array([c1,c2]).transpose(2,3,1,0)
```

```
In [159]: np.mean((X - tucker)**2)
```

```
Out[159]: 8.97580143460948
```

```
In [ ]: np.mean((X - b3)**2)
```

The cp composition resulted in a greater reduction of features (30 vs 38) and had a lower MSE (5 vs 9).