hw2q3_earth_movers_distance

February 16, 2018

```
In [1]: import matplotlib.pyplot as plt
        import numpy as np
        import pandas as pd
        import scipy.io
        from sklearn.metrics import euclidean_distances
```

```
In [2]: # load pt cloud. sample only 500 points in each pt cloud
        teapot = scipy.io.loadmat('teapot_ptcld.mat')
        pc1 = teapot['teapot'] # get XYZ coordinates
        pc1 = pc1[0:500,:]

        violin = scipy.io.loadmat('violin_ptcld.mat')
        pc2 = violin['violincase']
        pc2 = pc2[0:500,:]
```

```
In [3]: # Verify EMD with scipy fnction
        D = euclidean_distances(pc1, pc2)
        r, c = scipy.optimize.linear_sum_assignment(D)
        emd = D[r,c].sum()
        print(emd)

        # Test hungarian algorithm on small subset
        D_test = D[0:5,0:5]
        c_min = np.amin(D_test,axis=0)
        print(D_test)
        print(c_min)
        len(D_test)
```

```
26678.53632520494
[[ 54.80035406  56.23406786  55.14846459  53.93407919  54.49222339]
 [102.38935991 100.08554481 101.45725623 103.2249895  102.35071197]
 [ 49.83643266  49.71672305  50.04957095  50.26368402  50.20652196]
 [ 56.59239409  54.29657696  55.2144735   56.51930491  55.82456095]
 [ 51.49374156  51.25107785  51.62631362  52.30175913  51.96048157]]
[49.83643266 49.71672305 50.04957095 50.26368402 50.20652196]
```

```
Out[3]: 5
```

```
In [5]: mapping = D_test.copy()
        nrows = len(mapping)
        mask = mapping * 0

        # Step 1: Reduce rows by subtracting row mins from each row.
        for ind, row in enumerate(mapping):
            mapping[ind] -= row.min()

        # Step 2
        # Find a zero (Z) in the resulting matrix.  If there is
        # no starred zero in its row or column, star Z. Repeat for
        # each element in the matrix. Go to Step 3.
        for r, row in enumerate(mapping):
            for c, value in enumerate(row):
                if value == 0 and row_cover[r] == 0 and col_cover[c] == 0:
                    mask[r][c] = STARRED
                    row_cover[r] = 1
                    col_cover[c] = 1

            row_cover = np.zeros(nrows)
            col_cover = np.zeros(nrows)
```

```
        ---------------------------------------------------------------------------

        NameError                                 Traceback (most recent call last)

        <ipython-input-5-97c98ff65d2d> in <module>()
         13 for r, row in enumerate(mapping):
         14     for c, value in enumerate(row):
    ---> 15         if value == 0 and row_cover[r] == 0 and col_cover[c] == 0:
         16             mask[r][c] = STARRED
         17             row_cover[r] = 1


        NameError: name 'row_cover' is not defined
```