Assignment 3

March 29, 2018

1 Foundations of Data Mining: Assignment 3

Please complete all assignments in this notebook. You should submit this notebook, as well as a PDF version (See File > Download as).

Deadline: Thursday, March 29, 2018

```
In [9]: %matplotlib inline
        from preamble import *

        import sklearn.decomposition as deco
        import sklearn.manifold as manifold

        plt.rcParams['savefig.dpi'] = 100 # This controls the size of your figures
        # Comment out and restart notebook if you only want the last output of each cell.
        InteractiveShell.ast_node_interactivity = "none"#"all"
```

1.1 PCA and Isomap (5 Points, 1+2+2)

Apply PCA and Isomap to images of handwritten digits (see below). You may use sklearn.decomposition and sklearn.manifold.

1.1.1 a)

Compute the first two components of the data using PCA. Make a scatter plot of the data in the first two components of PCA indicating class with color.

1.1.2 b)

Compute an Isomap embedding with two components with nr_neighbors={5, 50, N-1} (three separate embeddings). For each of the Isomap embeddings, apply the function "align" (see below) with "ref_data" as your computed pca embedding and "data" as the isomap embedding. Show a scatter plot of each of the aligned isomap embeddings.

1.1.3 c)

Visually compare how well the classes are separated in the different scatter plots. What is the effect of changing the number of neighbors on the score computed in the alignment function? What does it mean if the score is zero? When do you expect the score to become zero and why?

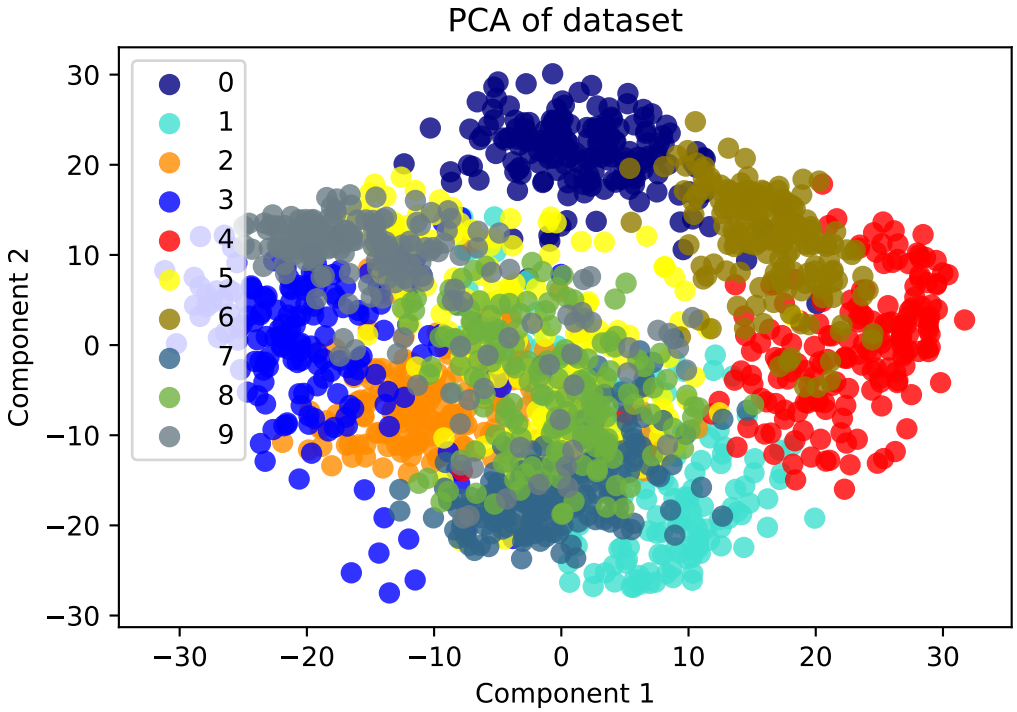
More neighbours seems to decrease the computed score. As the number of neighbours increases, the separation of the different classes seems to become more vague. For n=5 the classes seem decently seperated, while for n=50 and n=N-1 there is increasingly more overlap. If the score is zero, it means that the matrix norm of the transformed dataset minus the reference data (y) is zero for one of the 8 transformations. We expect this to happen if the transformed dataset is equal to the reference data for some transformation.

In [10]:
```
# Load the data set
from sklearn import datasets
digits = datasets.load_digits(n_class=10)
X = digits.data
y = digits.target
N=len(X)

# Align a data set with a reference data set minimizing l_1 error
# Returns aligned data set and alignment error
def align(ref_data, data):

    transformations = np.asarray([
        [[0,1],[1,0]],
        [[0,-1],[1,0]],
        [[0,1],[-1,0]],
        [[0,-1],[-1,0]],
        [[1,0],[0,1]],
        [[1,0],[0,-1]],
        [[-1,0],[0,1]],
        [[-1,0],[0,-1]]
    ])

    score = []
    for i in range(0,8):
        transf_data =   np.matmul(data, transformations[i])
        score.append(np.linalg.norm( transf_data - ref_data, ord=1) )
        #https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.linalg.nor

    idx = np.argmin(score)
    transf_data = np.matmul(data,transformations[idx])

    print("Aligned the data sets. Score is {0:10.1f}  ".format(score[idx]))

    return transf_data, score[idx]
```

In [11]:
```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
X_r = pca.fit(X).transform(X)
#print(pca.explained_variance_ratio_)
#print(pca.singular_values_)
```

```
#print(X_r)

print('1 a')
def printScatter1a():
    plt.figure()

    colors = ['navy', 'turquoise','darkorange', 'blue', 'red', 'yellow', 'xkcd:baby p
              'xkcd:ugly blue', 'xkcd:nasty green', 'xkcd:battleship grey'];

    for color, i in zip(colors, [0, 1, 2,3,4,5,6,7,8,9]):
        plt.scatter(X_r[y == i, 0], X_r[y == i, 1], color=color, alpha=.8, lw=2,
                    label=i);
    plt.legend(loc='best', shadow=False, scatterpoints=1);

    plt.title('PCA of dataset')
    plt.xlabel('Component 1')
    plt.ylabel('Component 2')

printScatter1a()
```

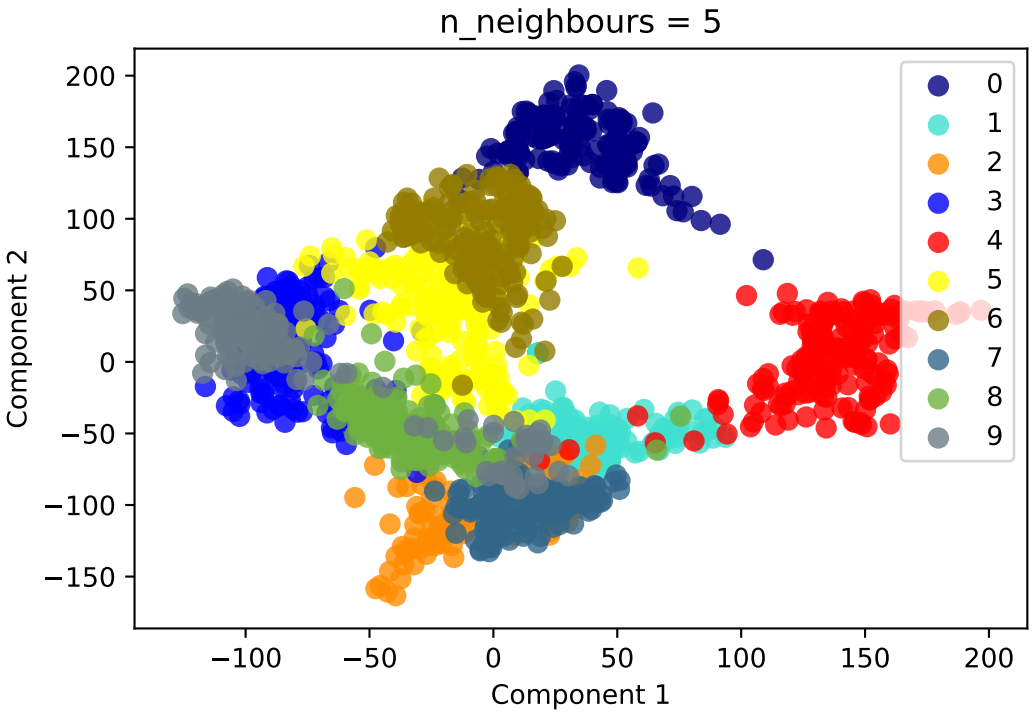
```
1 a
```

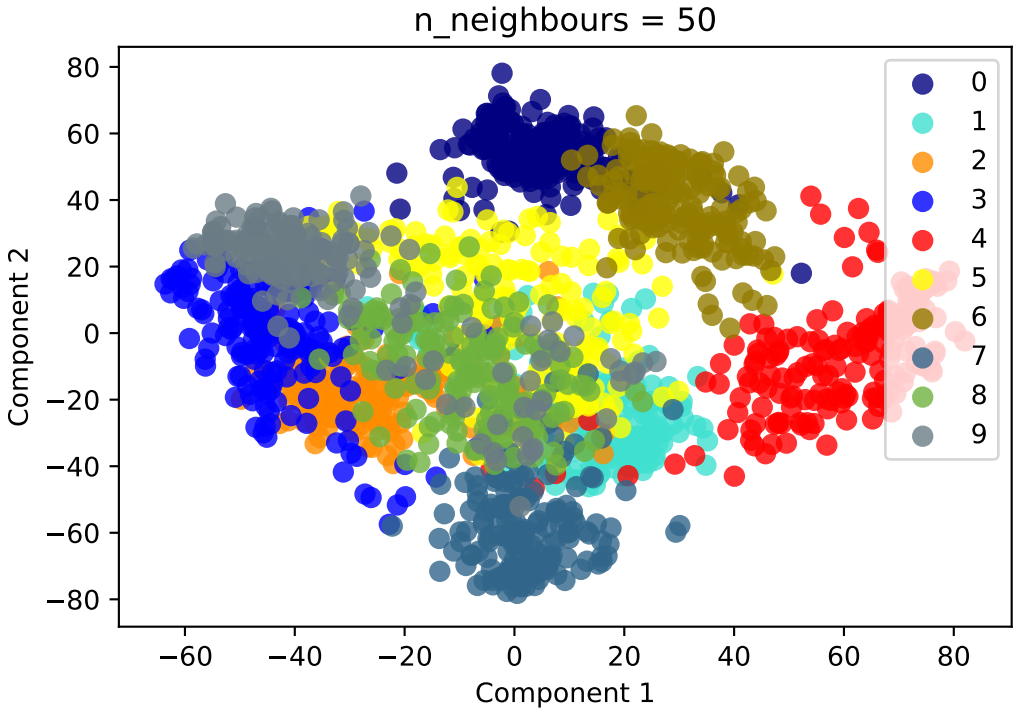


In [12]: # isomap embedding

```
from matplotlib import offsetbox
from matplotlib.colors import Colormap
from sklearn import (manifold, datasets, decomposition, ensemble,
                     discriminant_analysis, random_projection)

title = ['n_neighbours = 5', 'n_neighbours = 50', 'n_neighbours = N-1'];
values = [5,50,N-1];

print('1 b')
for n, title  in zip(values,title) :
    print(title);
    #print(n);
    X_iso = manifold.Isomap(n_neighbors=n, n_components=2).fit_transform(X)

    X_aligned,score = align(X_r,X_iso)

    plt.figure();
    plt.title(title);
    #plt.scatter(X_iso[:,0],X_iso[:,1])
    #plt.scatter(X_aligned[:,0],X_aligned[:,1])

    colors = ['navy', 'turquoise','darkorange', 'blue', 'red', 'yellow', 'xkcd:baby po
              'xkcd:ugly blue', 'xkcd:nasty green', 'xkcd:battleship grey'];

    for color, i in zip(colors, [0, 1, 2,3,4,5,6,7,8,9]):
        plt.scatter(X_aligned[y == i, 0], X_aligned[y == i, 1], color=color, alpha=.8
                    label=i);
    plt.legend(loc='best', shadow=False, scatterpoints=1);
    plt.xlabel('Component 1')
    plt.ylabel('Component 2')
    plt.title(title);
```

```
1 b
n_neighbours = 5
Aligned the data sets. Score is   102234.2
n_neighbours = 50
Aligned the data sets. Score is    31067.5
n_neighbours = N-1
Aligned the data sets. Score is        0.0
```


n_neighbours = 5
Component 2
Component 1
200
150
100
50
0
−50
−100
−150
−100
−50
0
50
100
150
200
0
1
2
3
4
5
6
7
8
9


n_neighbours = 50
Component 2
Component 1
80
60
40
20
0
−20
−40
−60
−80
−60
−40
−20
0
20
40
60
80
0
1
2
3
4
5
6
7
8
9