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 [2]:
```
%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 = "all"
```

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.

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.

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.

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?

In [1]:
```
# 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) )

    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]
```

Classical Multidimensional Scaling (6 Points, 1+2+2+1)

Show that for mean-centered data sets we can recover inner products using pairwise distance information only. This is used by the isomap embedding algorithm.

We are given all squared pairwise distances of an otherwise unknown point set $\mathbf{p}_1, \ldots, \mathbf{p}_n \in \mathbb{R}^d$, i.e., we are given for all $1 \leq i, j \leq n$ the values

$$d_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|^2.$$

We assume that the point set is mean-centered, that is

$$\sum_{i=1}^{n} \mathbf{p}_i = \vec{\mathbf{0}}.$$

(where $\vec{\mathbf{0}}$ is the vector of zeros)

In the following, $\langle \mathbf{p}_i, \mathbf{p}_j \rangle$ denotes the inner product of $\mathbf{p}_i$ and $\mathbf{p}_j$. Prove that the following holds true for mean-centered point sets:

$$-2\langle \mathbf{p}_i, \mathbf{p}_j \rangle = d_{ij} - \sum_{\ell=1}^{n} \frac{d_{i\ell}}{n} - \sum_{\ell=1}^{n} \frac{d_{j\ell}}{n} + \sum_{k=1}^{n} \sum_{\ell=1}^{n} \frac{d_{k\ell}}{n^2}$$

You may use the following steps in your derivation.

a)

Expand d_{ij} to yield an expression of $\langle \mathbf{p}_i, \mathbf{p}_j \rangle$, $\|\mathbf{p}_i\|^2$ and $\|\mathbf{p}_j\|^2$.

b)

Show that the following holds for any $\mathbf{q} \in \mathbb{R}^d$:

$$\sum_{1 \leq i \leq n} \langle \mathbf{p}_i, \mathbf{q} \rangle = 0$$

c)

Prove that

$$\|\mathbf{p}_i\|^2 = \sum_{\ell=1}^{n} \frac{d_{i\ell}}{n} - \sum_{k=1}^{n} \sum_{\ell=1}^{n} \frac{d_{k\ell}}{2n^2}$$

d)

Combine the steps in your proof.

Locality-sensitive hashing (4 Points, 2+1+1)

a)

Prove that if the Jaccard Similarity of two sets is 0, then minhashing always gives a correct estimate of the Jaccard similarity

b)

Let H be a family of (d_1, d_2, p_1, p_2)-locality-sensitive hash functions. Assume that $p_2 = 0$ and assume we have a total numer of m hash functions from this family available. Which combination of AND-constructions and OR-constructions should we use to maximally amplify the hash family?

(c)

Let H be a family of (d_1, d_2, p_1, p_2)-locality-sensitive hash functions. Assume that $p_2 = \frac{1}{n}$ and assume we have n data points $\mathbf{P}$ which are stored in a hash table using a randomly chosen function h from H. Given a query point $\mathbf{q}$, we retrieve the points in the hash bucket with index $h(\mathbf{q})$ to search for a point which has small distance to $\mathbf{q}$. Let X be a random variable that is equal to the size of the set

$$\{\mathbf{p} \in \mathbf{P} \ : \ h(\mathbf{p}) = h(\mathbf{q}) \ \wedge \ d(\mathbf{p}, \mathbf{q}) \geq d_2\}$$

which consists of the false positives of this query. Derive the expected number of false-positives $E[X]$.