COMPSCI 371D Homework 1

Problem 0 (3 points)

Part 1: Sets and Functions

Problem 1.1 (Exam Style)

Domain	Codomain	Map	Function?	Injection?	Surjection?	Biject
$\{1,2\}$	$\{a,b\}$	$\{(1,a),(1,b)\}$	No	No	No	N
$\{1,2\}$	$\{a,b\}$	$\{(1,a),(2,a)\}$	Yes	No	No	N
$\{1,2\}$	$\{a,b\}$	$\{(1,b),(2,a)\}$	Yes	Yes	Yes	Ye
$\{1,2\}$	$\{a,b,c\}$	$\{(2,a),(1,c)\}$	Yes	Yes	No	N
$\{1,2\}$	$\{b\}$	$\{(1,b),(2,b)\}$	Yes	No	Yes	N

Problem 1.2 (Exam Style)

$$n(a,b) = \binom{ab}{ab} + \binom{ab}{ab-1} + \ldots + \binom{ab}{1} = 2^{ab} - 1$$

$$n(3,3) = 511$$

$$n(2,4) = 255$$

$$n(5,3) = 32767$$

Problem 1.3 (Exam Style)

$$n(a,b) = \binom{b}{1}^a = b^a$$

$$n(3,3) = 27$$

$$n(2,4) = 16$$

$$n(5,3) = 243$$

Problem 1.4 (Exam Style)

$$a = b$$

$$n(a, b) = a!$$

$$n(4, 4) = 24$$

$$n(2, 4) = 0$$

$$n(5, 3) = 0$$

Problem 1.5 (Exam Style)

$$\text{distinct training sets of N samples} = \binom{m}{n} 2^n$$

$$\text{When} N = 5, M = 8 \text{ there are 1792 distinct training sets of N samples.}$$

Part 2: Fitting Banded Linear Transformations

Problem 2.1

In [2]:
```
from urllib.request import urlretrieve
from os import path as osp


def retrieve(file_name, semester='fall21', course='371d', homework=1):
    if osp.exists(file_name):
        print('Using previously downloaded file {}'.format(file_name))
    else:
        fmt = 'https://www2.cs.duke.edu/courses/{}/compsci{}/homework/{}/{}'
        url = fmt.format(semester, course, homework, file_name)
        urlretrieve(url, file_name)
        print('Downloaded file {}'.format(file_name))
```

In [3]:
```
import pickle


def read_data(file_name):
    retrieve(file_name)
    with open(file_name, 'rb') as file:
        d = pickle.load(file)
    return d
```

In [4]:
```
data = {data_set: read_data('{}.pkl'.format(data_set))
        for data_set in ('training', 'test')}
```

```
Using previously downloaded file training.pkl
Using previously downloaded file test.pkl
```

In [5]:
```
x_tr, y_tr = data['training']['x'], data['training']['y']
```

In [6]:
```
import numpy as np


def solve_system(u, v):
    return np.linalg.lstsq(u, v, rcond=None)[0]
```

In [7]:
```
h = solve_system(x_tr, y_tr)
```

In [8]:
```
def residual(h, x, y):
    diff = np.dot(x, h) - y
    r = np.linalg.norm(diff) / np.sqrt(x.size)
    return r
```

In [9]:
```
def diagonal_indicator(d, bandwidth):
    ind = np.zeros((d, d))
    for k in range(-bandwidth, bandwidth + 1):
        length = d - np.abs(k)
        ones = np.ones(length)
        ind += np.diag(ones, k=k)
    return ind.astype(bool)
```

In [10]:
```
def un_flatten_solution(h_flat, d, bandwidth):
    indicator = diagonal_indicator(d, bandwidth)
    h = np.zeros(d * d)
    h[indicator.ravel()] = h_flat
    h = np.reshape(h, (d, d))
    return h
```

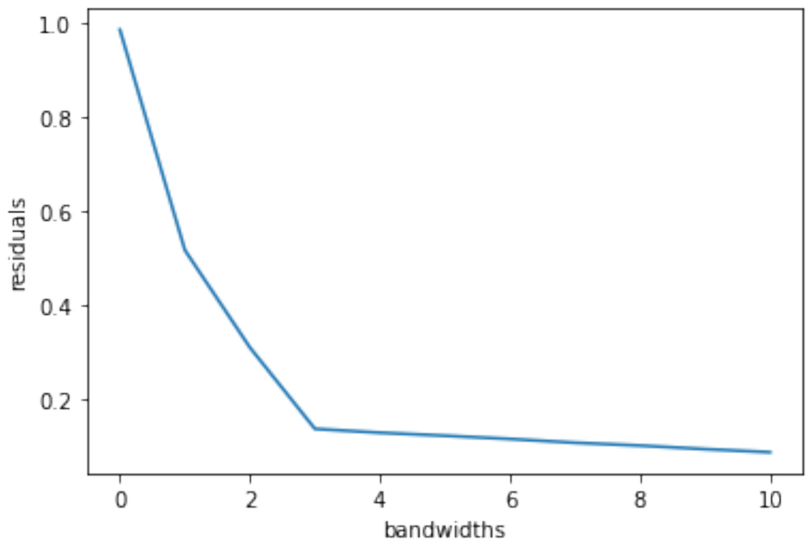
In [11]:
```
def flatten_system(x, y, bandwidth):
    y_flat = y.flatten()
    d = x.shape[1]
    A = np.kron(x, np.eye(d))
    flat_ind = diagonal_indicator(d,bandwidth).ravel()
    columns = np.arange(0,len(flat_ind))[flat_ind]
    A_c = A[:,columns]
    return A_c, y_flat
```

In [12]:
```
def fit_banded_matrix(x,y_o,bandwidth):
    A, y = flatten_system(x,y_o,bandwidth)
    h = un_flatten_solution(solve_system(A, y),x.shape[1],bandwidth)
    return h
```

In [13]:
```
from matplotlib import pyplot as plt
%matplotlib inline

residuals = []
for b in range(11):
    residuals.append(residual(fit_banded_matrix(x_tr, y_tr,b),x_tr, y_tr))

plt.plot(np.arange(0,11),residuals)
plt.xlabel("bandwidths")
plt.ylabel("residuals")
plt.show()
```



Problem 2.2 (Exam Style)

We can prove by contradiction that residuals must be weakly decreasing with respect to increasing bandwidths:

1. Assume towards contradiction that there exist bandwidths b and b' such that 0 <= b < b'; and that there exist residuals of prediction matrices with bandwidths of b and b' called r(b) and r(b') correspondingly, such that r(b) < r(b').
2. The hypothesis space of b (the set of banded dxd matrices with bandwidth b) forms a filtration in b, such that the hypothesis space of smaller b's are contained in the hypothesis space of larger b's.
3. If a prediction matrix with a bandwidth of b has a lower residual than that with b', it must also be contained in the hypothesis space of b', resulting in discovering a r(b') that is equal to r(b). Therefore, it is not possible for r(b') > r(b). It must be the case that r(b) >= r(b').

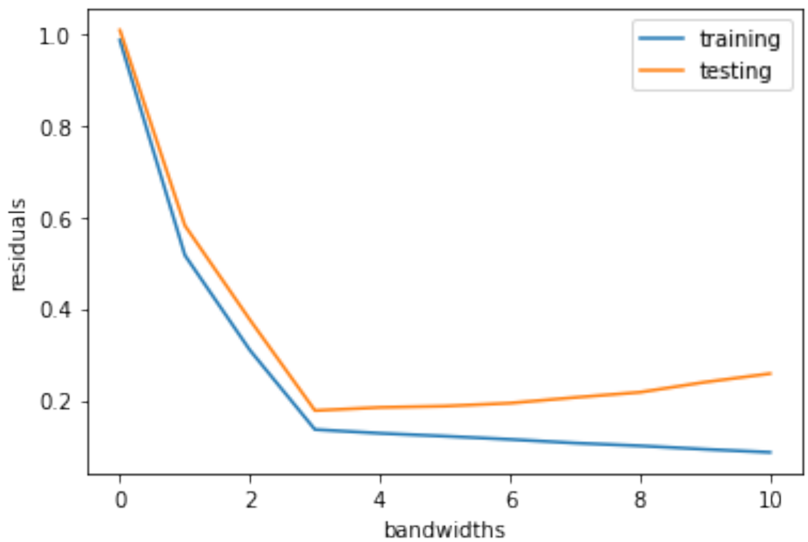
Part 3: Learning Banded Linear Transformations

Problem 3.1

```
In [14]: x_ts, y_ts = data['test']['x'], data['test']['y']

residuals_tr=[]
residuals_ts=[]
for b in range(11):
    H = fit_banded_matrix(x_tr, y_tr,b)
    residuals_tr.append(residual(H, x_tr, y_tr))
    residuals_ts.append(residual(H, x_ts, y_ts))

plt.plot(np.arange(0,11),residuals_tr)
plt.plot(np.arange(0,11),residuals_ts)
plt.xlabel("bandwidths")
plt.ylabel("residuals")
plt.legend(["training", "testing"])
plt.show()
```



Problem 3.2 (Exam Style)

Bandwidth of 3 would be the best. While the residuals for the training set keep decreasing for increased bandwidths beyond 3, the residuals for the testing set start to increase after a bandwidth of 3. At a bandwidths of 3, the prediction matrix yields the lowest residuals for the testing set, which means that 3 would be the best bandwidth for this case.