COMPSCI 371D Homework 1

Jihyeon Je (jj271), Tim Ho (th265), Rahul Prakash (rp221)

Problem 0 (3 points)

Part 1: Sets and Functions

Problem 1.1 (Exam Style)

Domain	Codomain	Map	Function?	Injection?	Surjection?	Bijection?	None of these
$\{1, 2\}$	$\{a, b\}$	$\{(1, a), (1, b)\}$					*yes*
$\{1, 2\}$	$\{a, b\}$	$\{(1, a), (2, a)\}$	*yes*				
$\{1, 2\}$	$\{a, b\}$	$\{(1, b), (2, a)\}$	*yes*	*yes*	*yes*	*yes*	
$\{1, 2\}$	$\{a, b, c\}$	$\{(2, a), (1, c)\}$	*yes*	*yes*			
$\{1, 2\}$	$\{b\}$	$\{(1, b), (2, b)\}$	*yes*		*yes*		

Problem 1.2 (Exam Style)

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

numerical examples:

$$n(3,3) = \binom{9}{1} + \binom{9}{2} + \binom{9}{3} + \binom{9}{4} + \binom{9}{5} + \binom{9}{6} + \binom{9}{7} + \binom{9}{8} + \binom{9}{9} = 511$$

$$n(2,4) = \binom{8}{1} + \binom{8}{2} + \binom{8}{3} + \binom{8}{4} + \binom{8}{5} + \binom{8}{6} + \binom{8}{7} + \binom{8}{8} = 255$$

$$n(5,3) = \binom{15}{1} + \binom{15}{2} + \binom{15}{3} + \binom{15}{4} + \binom{15}{5} + \binom{15}{6} + \binom{15}{7} + \binom{15}{8} + \binom{15}{9} + \binom{15}{10} + \binom{15}{11} + \binom{15}{12} + \binom{15}{13} + \binom{15}{14} + \binom{15}{15} = 32767$$

Problem 1.3 (Exam Style)

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

numerical examples:

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

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

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

Problem 1.4 (Exam Style)

iff a = b,

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

if a ≠ b, then

$$n(a, b) = 0$$

numerical examples:

$$n(4, 4) = 4! = 24$$
$$n(2, 4) = 0$$
$$n(5, 3) = 0$$

Problem 1.5 (Exam Style)

total number of combinations of messages is:

$$\binom{M}{N}$$

each message can be either labeled true or false, so all possible combinations of true/false within the training set is:

$$2^N$$

Therefore, the number of total distinct training sets of N samples from M messages is:

$$\binom{M}{N} * 2^N$$

Sanity check:

$$M = 8, N = 5$$
$$\binom{8}{5} * 2^5 = 1792$$

Part 2: Fitting Banded Linear Transformations

Problem 2.1

In [1]:
```
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 [2]:
```
import pickle


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

In [3]:
```
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 [4]:
```
x_tr, y_tr = data['training']['x'], data['training']['y']
```

In [5]:
```
import numpy as np


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

In [6]:
```
def residual(h, x, y):
    diff = np.dot(x, h) - y
    r = np.linalg.norm(diff) / np.sqrt(x.size)
    return r
```

In [7]:
```
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)
```

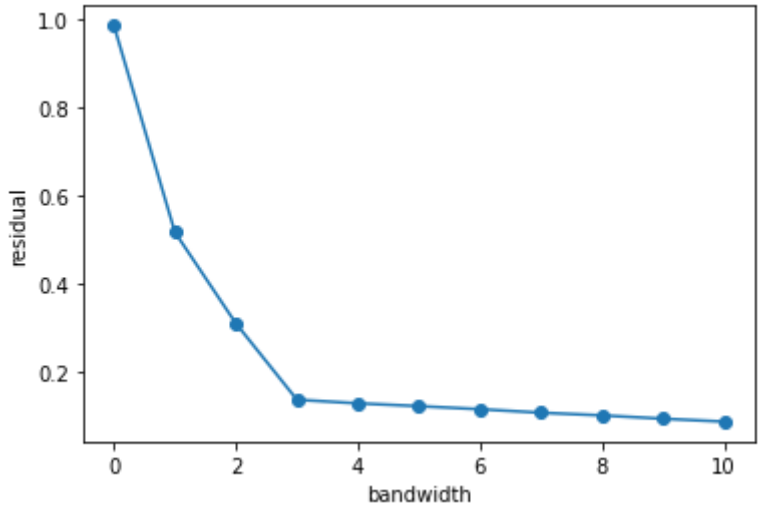
In [8]:
```
def un_flatten_solution(h_flat, d, bandwidth):
    indicator = diagonal_indicator(d, bandwidth)
    h = np.zeros(d * d)
    h[indicator.ravel()] = h_flat
    h = np.reshape(h, (d, d))
    return h
```

In [9]:
```
#defined function
def flatten_system(x,y,bandwidth):
    y_flat = y.flatten() #small y

    full = np.kron(x, np.eye(100))
    diagindc = diagonal_indicator(100, bandwidth)
    idx = diagindc.flatten()
    A = full[:,idx]


    return A, y_flat
```

In [10]:
```
def fit_banded_matrix(x, y, d, bandwidth):
    A, y_flat = flatten_system(x, y, bandwidth)
    h = solve_system(A,y_flat)
    #print(h.shape)
    H = un_flatten_solution(h, d, bandwidth)
    return H
```

In [11]:
```
# calculate residuals for all bandwidth values
totlist = []
for i in range(0,11):

    bestfit = fit_banded_matrix(x_tr, y_tr, 100, i)
    res = residual(bestfit, x_tr, y_tr)
    totlist.append(res)
```

In [12]:
```
# plot the result
import matplotlib.pyplot as plt
x = totlist
y = [0,1,2,3,4,5,6,7,8,9,10]
plt.plot(y,x,'-o')
plt.xlabel('bandwidth')
plt.ylabel('residual')
plt.show()
```



Problem 2.2 (Exam Style)

From the plot above, we see that the plot is weakly decreasing. The monotonicity of the training risk could be proven by contradiction.

1. Assume that the residuals are monotonically increasing as bandwidth increases.
2. The hypothesis space $\mathcal{H}_d(b)$ form a filtration in b, therefore
$$\mathcal{H}_d(0) \subset \mathcal{H}_d(1) \subset \ldots \subset \mathcal{H}_d(d-1)$$
3. $\mathcal{H}_d(n)$ is a subspace of $\mathcal{H}_d(n+1)$ where n is a integer $0 \leq n < d$
4. The residual of $\mathcal{H}_d(n+1)$ must be greater than the residual of $\mathcal{H}_d(n)$ according to 1.
5. Statement 4 contradicts statement 1 as the best fit H $\in \mathcal{H}_d(n)$ is also $\in \mathcal{H}_d(n+1)$ by statement 3.
6. Therefore, the residual obtained from $\mathcal{H}_d(n+1)$ must be less than or equal to the residual obtained from $\mathcal{H}_d(n)$. Since this contradicts statement 1, we prove by contradiction that the residuals are monotonically non increasing.

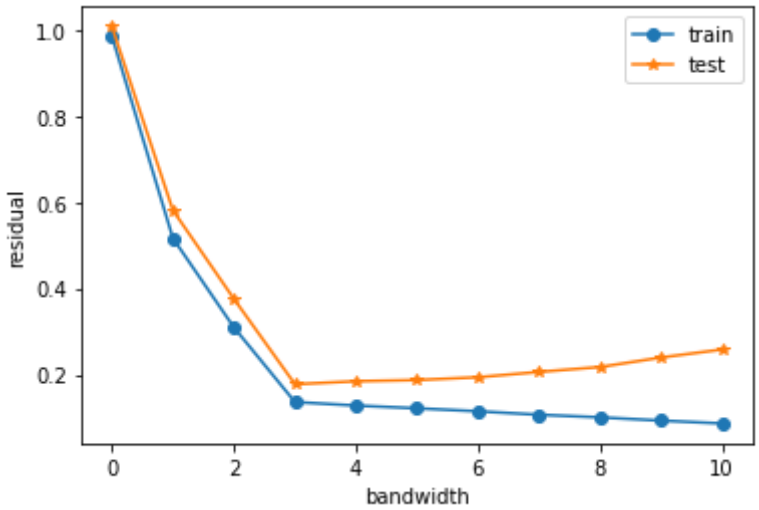
Part 3: Learning Banded Linear Transformations

Problem 3.1

In [13]:
```
x_ts, y_ts = data['test']['x'], data['test']['y']
```

In [14]:
```
# calculate residuals for all bandwidth values
trainres = []
testres = []
for i in range(0,11):

    bestfit = fit_banded_matrix(x_tr, y_tr, 100, i)
    tr = residual(bestfit, x_tr, y_tr)
    trainres.append(tr)
    ts = residual(bestfit, x_ts, y_ts)
    testres.append(ts)
```

In [15]:
```
# plot the result
import matplotlib.pyplot as plt
y = [0,1,2,3,4,5,6,7,8,9,10]
plt.plot(y,trainres,'-o',  label= 'train')
plt.plot(y,testres,'-*',  label= 'test')
plt.xlabel('bandwidth')
plt.ylabel('residual')
plt.legend()
plt.show()
```



Problem 3.2 (Exam Style)

A bandwidth of 3 was likely used to generate the data since this value minimizes the residual for the test set. The test residuals are not monotonic because the residual values decrease until bandwidth of 3 then starts to increase past the value. This increase is likely due to overfitting on the training set. At this point H* starts to attempt to model the exact noise present in the training set. When the added random noise is changed in the test set, this results in an increased error (residual).