Caltech Assignment

Rajat Goel

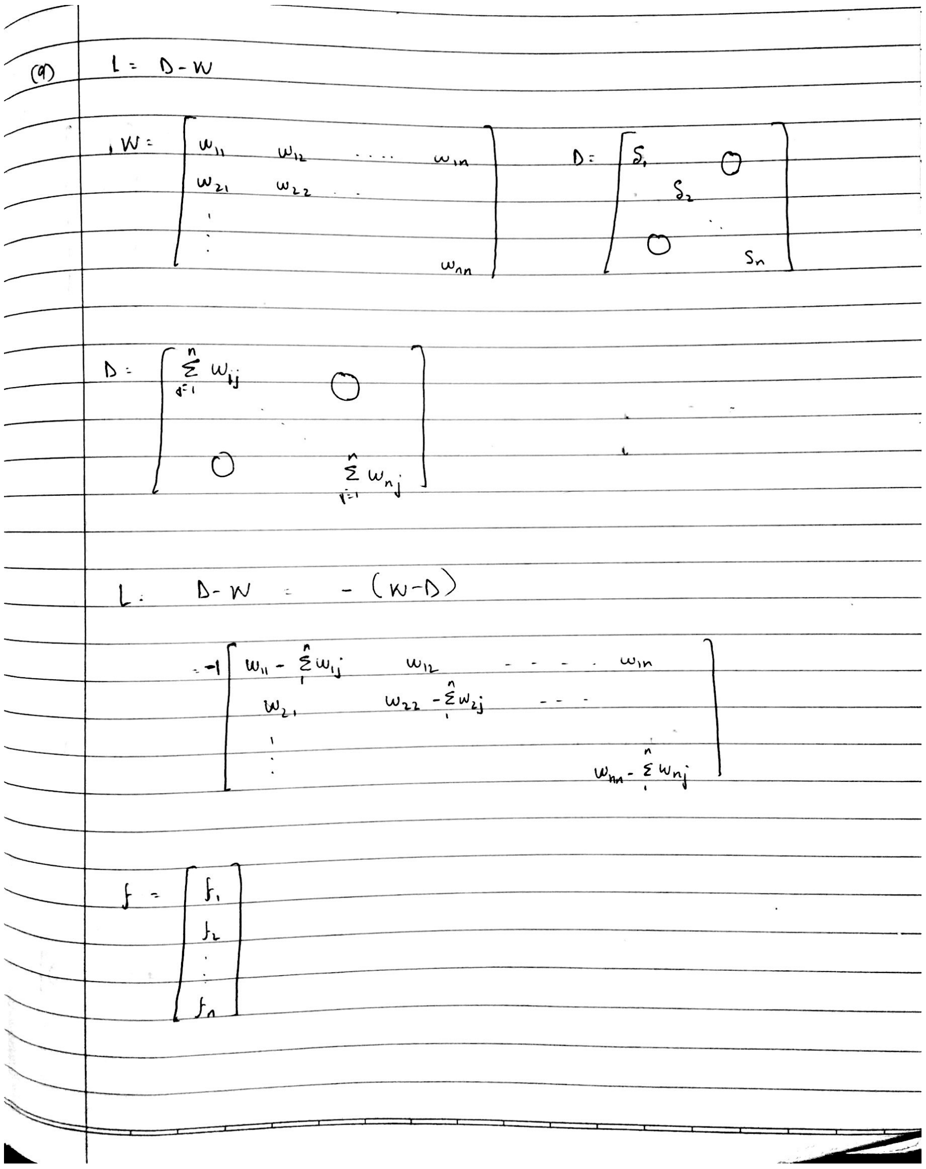
November 17,2017

Contents

All my codes are available in this repository: `https://github.com/R1j1t/Caltech-Assignment`

0.1 Question-1

0.1.1 Part-a (i)



(a) $L = D - W$

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & \\ \vdots & & & \\ & & & w_{nn} \end{bmatrix} \quad D = \begin{bmatrix} \delta_1 & & 0 \\ & \delta_2 & \\ & & \ddots & \\ 0 & & & \delta_n \end{bmatrix}$$

$$D = \begin{bmatrix} \sum_{j=1}^{n} w_{1j} & & 0 \\ & \ddots & \\ 0 & & \sum_{j=1}^{n} w_{nj} \end{bmatrix}$$

$$L = D - W = -(W - D)$$

$$= -1 \begin{bmatrix} w_{11} - \sum_{1}^{n} w_{1j} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} - \sum_{1}^{n} w_{2j} & \cdots & \\ \vdots & & & \\ & & & w_{nn} - \sum_{1}^{n} w_{nj} \end{bmatrix}$$

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}$$

$$\langle f, Lf \rangle = f^T(Lf)$$

$$= [f_1 \;\; f_2 \;\; \dots \;\; f_n] \, (Lf)$$

$$Lf = - \begin{bmatrix} w_{11} - \sum_1^n w_{1j} & w_{12} & & w_{1n} \\ w_{21} & & & \\ \vdots & & & \\ \vdots & & & \\ w_{n1} & & & w_{nn} - \sum_1^n w_{nj} \end{bmatrix}_{n \times n} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_n \end{bmatrix}_{n \times 1}$$

$$= - \begin{bmatrix} f_1 w_{11} - f_1 \sum w_{1j} + f_2 w_{12} - \dots \quad w_{nn} f_n \\ w_{21} f_1 + f_2 w_{22} - f_2 \sum_1^n w_{2j} \quad \dots \\ \vdots \\ \vdots \\ f_1 w_{n1} + f_2 w_{n2} \quad \dots \quad + f_n w_{nn} - f_n \sum_1^n w_{nj} \end{bmatrix}_{n \times 1}$$

$$\langle f, Lf \rangle = [f_1 \;\; f_2 \;\; \dots \;\; f_n] \, (Lf)$$

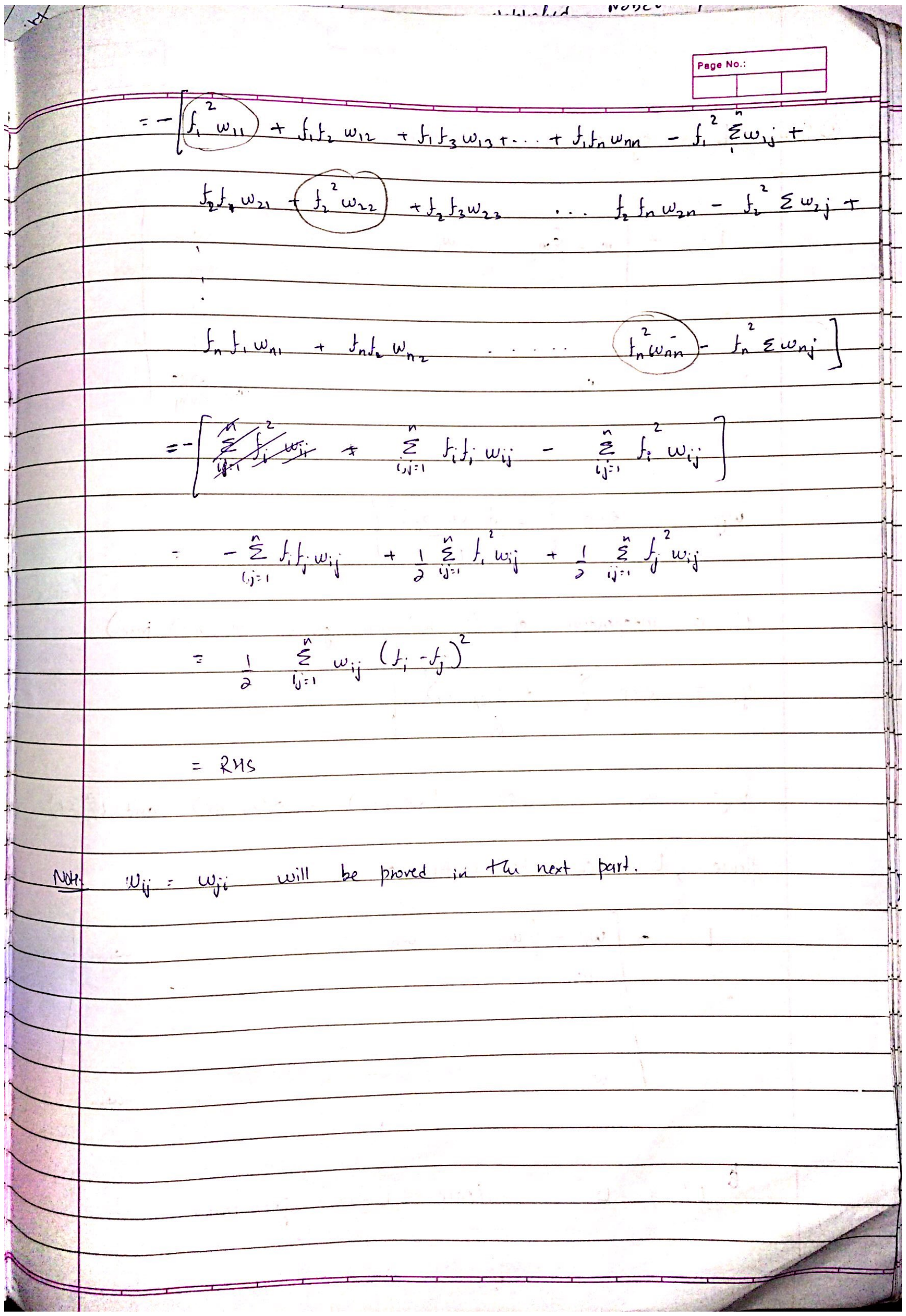
$$= - \Big[f_1 \big(f_1 w_{11} - f_1 \sum w_{1j} + f_2 w_{12} \quad \dots \quad f_n w_{nn} \big) +$$

$$f_2 \big(w_{21} f_1 + f_2 w_{22} - f_2 \sum w_{2j} \quad \dots \big) +$$

$$\vdots$$

$$+$$

$$f_n \big(f_1 w_{n1} + f_2 w_{n2} \quad \dots \quad + f_n w_{nn} - f_n \sum w_{nj} \big) \Big]$$



Page No.:

$$= -\Big[\boxed{f_1^2 w_{11}} + f_1 f_2 w_{12} + f_1 f_3 w_{13} + \dots + f_1 f_n w_{1n} - f_1^2 \sum_1^n w_{1j} +$$

$$f_2 f_1 w_{21} + \boxed{f_2^2 w_{22}} + f_2 f_3 w_{23} \quad \dots \quad f_2 f_n w_{2n} - f_2^2 \sum w_{2j} +$$

$$\vdots$$

$$f_n f_1 w_{n1} + f_n f_2 w_{n2} \quad \dots\dots \quad \boxed{f_n^2 w_{nn}} - f_n^2 \sum w_{nj}\Big]$$

$$= -\Big[\cancel{\sum_{i=1}^{n} f_i^2 w_{ii}} + \sum_{i,j=1}^{n} f_i f_j w_{ij} - \sum_{i,j=1}^{n} f_i^2 w_{ij}\Big]$$

$$= -\sum_{i,j=1}^{n} f_i f_j w_{ij} + \frac{1}{2}\sum_{i,j=1}^{n} f_i^2 w_{ij} + \frac{1}{2}\sum_{i,j=1}^{n} f_j^2 w_{ij}$$

$$= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$$

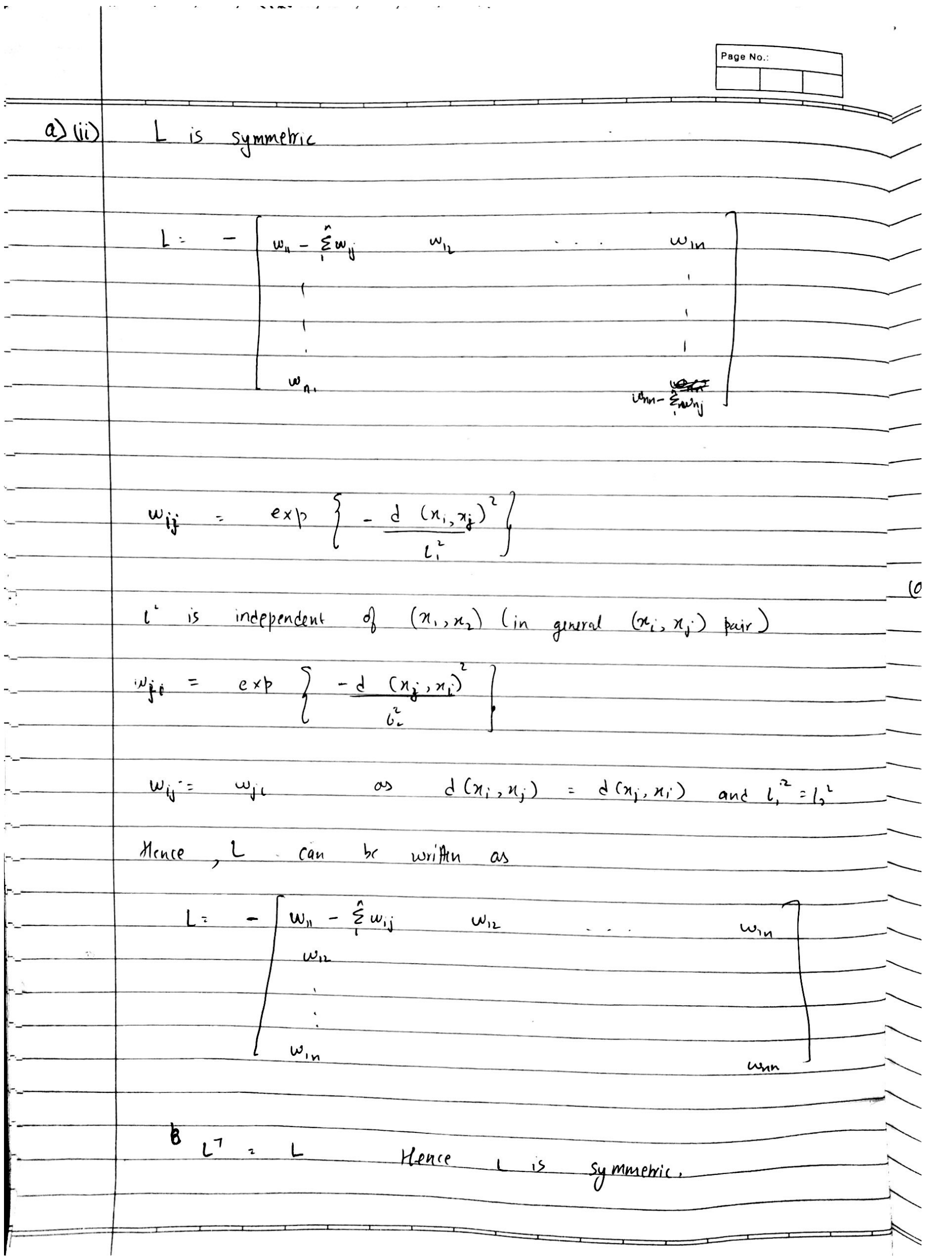
$$= \text{RHS}$$

Note: $w_{ij} = w_{ji}$ will be proved in the next part.

As w_{ij} is symmetric (proved in the next part) we can say

$$\sum_{i,j}^{n} f_i^2 w_{ij} = \frac{1}{2}\sum_{i,j}^{n} f_i^2 w_{ij} + \frac{1}{2}\sum_{i,j}^{n} f_j^2 w_{ij}$$

0.1.2 Part-a (ii)



Page No.:

a) (ii) L is symmetric

$$L = - \begin{bmatrix} w_{11} - \sum_{1}^{n} w_{j} & w_{12} & \cdots & w_{1n} \\ \vdots & & & \vdots \\ w_{n1} & & & w_{nn} - \sum_{1} w_{nj} \end{bmatrix}$$

$$w_{ij} = \exp\left\{ - \frac{d(x_1, x_j)^2}{l_1^2} \right\}$$

l^2 is independent of (x_1, x_2) (in general (x_i, x_j) pair)

$$w_{ji} = \exp\left\{ - \frac{d(x_j, x_i)^2}{l_2^2} \right\}$$

$w_{ij} = w_{ji}$ as $d(x_i, x_j) = d(x_j, x_i)$ and $l_1^2 = l_2^2$

Hence, L can be written as

$$L = - \begin{bmatrix} w_{11} - \sum_{1}^{n} w_{1j} & w_{12} & \cdots & w_{1n} \\ w_{12} & & & \\ \vdots & & & \\ w_{1n} & & & w_{nn} \end{bmatrix}$$

$L^T = L$ Hence L is symmetric.

Page No.:

from the defination of semi-positive

$z^T L z \geq 0$

and from part (a) - (i) we have

$\langle f, Lf \rangle = f^T L f = \frac{1}{2} \sum w_{ij} (f_i - f_j)^2$

for ~~no~~ $f \in \mathbb{R}^n - \{0\}$

$f^T L f = 0$ for $f_i = f_j$

(a) (iii)

0.1.3 Part-a (iii)

To show that for Eigen value = 0 has geometric multiplicity 1 iff graph as 1 connected components (a, b). We will use the equality from the previous part of this question f'Lf = $\frac{1}{2}\sum_{i,j}^{n} f_j^2 w_{ij}$

So we have for λ =0,

$$(A - \lambda * I)v = 0 \tag{1}$$
$$(A - \bar{0})v = 0 \tag{2}$$
$$Av = 0 \tag{3}$$
$$v'Av = 0 \tag{4}$$
$$\frac{1}{2}\sum_{i,j=0}^{n} w_{ij}(v_i - v_j)^2 = 0 \tag{5}$$

For 1 connected components $w_{ab} \neq 0 \Rightarrow v_a = v_b$ and $v_i, v_j \neq 0$ (for i,j $\neq$ a,b). From this we get the geometric multiplicity 1.

0.1.4 Part-a (iv)

Generalizing this fact for k connected components we say matrix L can be converted to block diagonal form as shown

$$\begin{pmatrix} L_1 & 0 & 0 & .... & 0 \\ 0 & L_2 & 0 & .... & 0 \\ & & & & \\ 0 & 0 & 0 & .... & L_n \end{pmatrix} \tag{6}$$

using $v_i' L_i v = 0$ and each l_i has 1 connected components. In total we have k connected components for $\lambda = 0$. Hence geometric multiplicity is k for k connected components.

0.1.5 Part-b (i)

In this problem we had to find the number of connected components, given the graph with weight edges. To solve for all the possible case, I have written a python code given below and with it, I found the connected sets to be (x_1, x_2),(x_1, x_3),(x_2, x_3),(x_1, x_2, x_3).

The python code is:

```python
import itertools


a=['x1','x2','x3','x4','x5']
b=[]
c={}
```

```python
paths=[]
connectedComponents=[]

for j in itertools.permutations(a,2):
    if len(set(j).intersection(('x1', 'x2'))) == len(j):
        c[j]=1
    elif  len(set(j).intersection(('x1', 'x3'))) == len(j):
        c[j]=1
    elif len(set(j).intersection(('x2', 'x3'))) == len(j):
        c[j]=1
    else:
        c[j]=0

for i in range(2,len(a)+1):
    for j in itertools.combinations(a,i):
        b.append(j)

for i in b:
    if len(i)==2:
       if c[i]==1:
           paths.append(i)
           connectedComponents.append(i)

    else:
        product = 1
        count0=0

        for k in itertools.combinations(i,2):
            product = c[k]*product
            if c[k] == 0:
                count0 +=1

        if product == 1:
            paths.append(i)
        if count0 <=1 :
            connectedComponents.append(i)

print('Connected Components are:')
print(connectedComponents)
```

Hence using the results from above code, we can say x_1, x_2 and x_1, x_2, x_3 are connected components and we have a total of 4 connected components for the given weighted edge graph.

0.1.6 Part-b (ii)

Eigen Values

$$A \in \mathbb{C}^{n_1 * n_2} \tag{7}$$

$$A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} where; A_1 \in \mathbb{C}^{n_1 * n_1}; A_2 \in \mathbb{C}^{n_2 * n_2}$$

(8)

λ of $A_1 = \lambda_1^1....\lambda_n^1$ λ of $A_2 = \lambda_1^2....\lambda_n^2$
To find eigen values of Matrix A,

$$\begin{pmatrix} A_1 - \lambda * I & 0 \\ 0 & A_2 - \lambda * I \end{pmatrix} = 0 \tag{9}$$

$$= det(A_1 - \lambda * I_{n1}) * det(A_2 - \lambda * I_{n2})$$

$$= (\lambda - \lambda_1^1)....(\lambda - \lambda_{n1}^1)(\lambda - \lambda_1^2)....(\lambda - \lambda_{n2}^1)$$

Hence the eigen values for A are $\lambda_1^1, ..., \lambda_{n1}^1, \lambda_1^2 ...., \lambda_{n2}^2$

Eigen Vectors

$$\begin{pmatrix} A_1 - \lambda * I & 0 \\ 0 & A_2 - \lambda * I \end{pmatrix} \text{V=0 where; V is the eigen vector vector} \tag{10}$$

$$\begin{pmatrix} A_1 - \lambda * I & 0 \\ 0 & A_2 - \lambda * I \end{pmatrix} \quad \begin{pmatrix} w_1^{n1} \\ w_2^{n2} \end{pmatrix} = \overline{0} \tag{11}$$

$$(A_1 - \lambda * I)w_1^{n1} = 0$$

for

$$w_1^{n1} = v_1^1, v_2^1, ..., v_{n1}^1 \quad and \quad \lambda = \lambda_1^1, ..., \lambda_{n1}^1$$

Now the question arises for $(A_2 - \lambda * I)w_2^{n2}$.

Now making an assumption that $\lambda \quad of A_1 and A_2$ are not same of A_1 and A_2 are same then we will have $w_2^{n2} = \overline{0}$. With this we can conclude that eigen vectors of A are

$$V = \begin{pmatrix} v_1^1 & v_2^1 & .... & v_{n1}^1 & 0 & 0 & .... & 0 \\ 0 & 0 & .... & 0 & v_1^2 & v_2^2 & .... & v_{n2}^2 \end{pmatrix} \tag{12}$$

Generalising for large of blocks

We can generalise this as follows:

$$V = \begin{pmatrix} V_1 & 0 & 0 & .... & 0 \\ 0 & V_2 & 0 & .... & 0 \\ \\ \\ 0 & 0 & 0 & .... & V_n \end{pmatrix} \tag{13}$$

where V_n = eigen vectors for nth *block*

0.2 Question-2

0.2.1 Part-a

The solution code for this part below:

```
function [L, Lsym, Lrw, W] = Lmatrix(X,l)

if l<0
    fprintf('l should be greater than zero');
    return
end

n=size(X,1);
W=zeros(n);
D=zeros(n);

for i= 1:n

    for j=1:n

        if i<j
            dist = norm(X(i,:)-X(j,:));
            wij = exp(-(dist^2)/(l^2));
           W(i,j)=wij;
           W(j,i)=wij;

        elseif i==j
            dist = norm(X(i,:)-X(j,:));
           W(i,j) = exp(-(dist^2)/(l^2));

        end
    end
end


for i=1:n
    D(i,i)=sum(W(i,:));
end

L=D-W;
Lsym = (D^(-0.5))*L*(D^(-0.5));
Lrw = (D^(-1))*L;
end
```

With this we will get graph Laplacian L, L_{sym}, L_{rw} for the given data set X and choice of l.

0.2.2 Part-b (i)

As given in Question-1,part-b of the assignment, for $\lambda = 0$ we have k connected components if the geometric multiplicity of the graph laplacian is k. Hence we will find an provided an initial guess of l and then iterated over different values of l to kind the 1st occurrence, of eigen value = 0. With l known for λ =0 we generate the graph laplacians and see the number of blocks in these matrix. For $data2.mat$, number of block in $L/L_{sym}/L_{rw} = 6$. Hence the number of clusters from eigen values is estimated to be 6.

For L

Clustering is very sensitive for L, for variation in l ($\lambda = 0$) the number data points in each clusters vary drastically from 330 (l = 0.0018) data points in each cluster to $\approx$ 170 (l = 0.0014).
Same goes for larger values of l($\approx$ 16).

For L_sym

Clustering is easily visible for values of l in the range ($\lambda = 0$) 4.046 ¡ l ¡ 10.816. Clustering also remained uniform over this range of l. For some smaller values of l (like 0.0013) the graph laplacian was not converted to perfect block diagonal form (but the elements had values of the order of -100). The clustering of data set was erratic.

For L_rw

Clustering is easily visible for values of l in the range ($\lambda = 0$) 1.53 . Clustering also remained uniform over this range of l. For some smaller values of l (like 0.001) the graph laplacian was not converted to perfect block diagonal form (but the elements had values of the order of -200). The data was classified into 1 cluster only for small l.

Qualitatively speaking, variation of number of data point in a clustering varied minutely for the right selection of l.

0.2.3 Part-b (ii)

For L

Clustering is not being performed for L. **Almost all** the data points were grouped in to 1 group.

For L_sym

As mentioned above clustering obtained by self-tuning weights is very different from that obtained in part (i). In term of distributions more than 200 data points were clustered into 1 cluster (for K=10).

For L_rw

The clustering obtained by self-tuning weights is very different from that obtained in part (i). In term of distributions more than 200 data points were clustered into 1 cluster (for K=10), whereas in part (i) 60-70 were clustered in 1 group.

0.2.4 Part-b (iii)

I am using self tuning weights for the below analysis.

For X2 and K=10

For L

With more noise, the performance of L remained same, i.e. does not classify at all.

For L_sym

For L_sym the performance doesn't degrade much, and the number of clusters were also easily identified.

for L_rw

For L_sym the performance *changed* based on number of data points in a cluster. The number increased from somewhere around 200 to 250 in one of the clusters.

For X3 and K=10

For L

With even more noise, the clustering performed as expected and all the data points were grouped as 1.

For L_sym

For L_sym the performance doesn't degrade much, and the number of clusters were also easily identified. But this time compared to the previous case maximum of data points in 1 cluster increased marginally.

for L_rw

For L_sym the performance changed, this comment is being made on the number of data points in a cluster. The number decreased from somewhere around 200 to 150 in one of the clusters. Some noise was also observed in the block diagonal form.

0.2.5 Part-b (iv)

I didn't quite understand this part and hence wasn't able to attempt.

0.3 Question-3

0.3.1 Part-(i)

The graph Laplacian chosen is $L_s ym$ and value of K is varied from 32 to 1024 (in power 2). The results are as follows.

For K=32,64,128, the data set distribution was C_1 *size* $\approx$ 100, 100, 700 respectively. C_2 *size* $\approx$ was 1300, 500 and 1000 respectively. For C_3 *size* it varied from $\approx$ 1600, 2500, 1400. Below I am mentioning all the data in a tabular form.

Variation of Cluster size (approx) with value of K			
Value of K	C_1 *size*	C_2 *size*	C_3 *size*
32	100	1300	1600
64	100	500	2500
128	700	1000	1400
256	900	1000	1100
512	950	1000	1135
1024	950	970	1165

0.3.2 Part-(ii)

Based on the above table we can say that, 1 cluster is easier to identify than other two. But this difference is not very large. The reason I think is, because spectral clustering works on the idea of similarity [1][2].

Referring to the this idea we say that Spectral clustering for '4' and '9' would be quite same and complex when compared with '1'. Hence it would be easier to identify '1' than other 2 example types because of its simpler structure.

Bibliography

[1] Spectral clustering lecture. `https://www.cs.cmu.edu/~aarti/Class/10701/slides/Lecture21_2.pdf`. Accessed: 2017-11-16.

[2] Ulrike von Luxburg. A tutorial on spectral clustering. Statistics and Computing, 17 (4), 2007.