1 Meeting Summary 07/01/2020

1.1 Gaussian Similarity Function

We started by discussing the role of the scaling factor σ in the fully connected similarity graph with exponentially decaying weights ω_{ij} for $i, j = 1, ..., n$ computed using the Gaussian similarity formula.

$$\omega_{ij} = \exp\Big(-\frac{||x_i - x_j||^2}{2\sigma^2}\Big)$$

For computing the pairwise Euclidean distance $||x_i - x_j||$ for the entire data set, the first step would be to construct the **distance matrix**. The distance matrix is an n x n symmetric matrix composed by the distance measures for each pair of cases in the data set, hence with a zero valued - main diagonal.

For instance, let $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})^T$ where the $i : th$ element is the vector $\mathbf{x_i} = (x_{i1}, x_{i2}, ..., x_{iD})$, where D denotes the dimension of the vector for $i = 1, ..., n$ and $D \in \mathbb{N}_+$. The distance matrix will take the following form:

$$\begin{pmatrix} 0 & & & & \\ d_{2,1} & 0 & & & \\ d_{3,1} & d_{3,2} & 0 & & \\ \vdots & & & & \\ d_{n,1} & d_{n,2} & \ldots & d_{n,n} & 0 \end{pmatrix}$$

Where d_{ij} represents the distance measure between the i:th and j:th element in our data set. If our data was on a D-dimensional space, and we were using the **Euclidean Distance** then

$$d_{i,j} = \sqrt{(x_{i1} - x_{j1})^2 + ... + (x_{iD} - x_{jD})^2}$$

, and by symmetry $d_{i,j} = d_{j,i}$.

If the separation of two points i, j is smaller than σ then their connection will have a significant weight ω_{ij} as both points will reside in the same neighborhood. However, if the distance was bigger than σ, then our weight will have a very small value, meaning that σ will somehow define the extend of our neighborhood. The higher the value of σ is the stronger connection weights we will have and the smaller the value of σ is, the weaker the connections between points will be. We can also think about σ as the radius of the circular cluster.

We want to choose a value of σ which will allow us only to connect neighboring points but not too big, as this will allow connections between inter-component points. Ideally we will like to get a natural boundary, that will allow us to easily separate the components in the graph.

1.2 Gaussian Similarity Function: Imbalanced data

How to choose σ when dealing with unbalanced data? (adapted σ)-Not to worry too much about this now.

1.3 Unnormalized graph Laplacian

The unnormalized graph Laplacian is defined as

$$L = D - W$$

The components of the graph Laplacian are:

- **Diagonalized degree matrix** D: n x n diagonal matrix. In the case of an unweighted graph the diagonal elements represent the number of connections to the rest of nodes in the network, and for a weighted graph each element will be the sum of connective weights per data points. Hence, each diagonal element will be:

$$D_{ii} = \sum_{j=1}^{n} \omega_{ij}$$

  This matrix will gives us an idea of how connected each data point is to the rest of the network.

- **Adjacency matrix** W: is the matrix representation of the graph, with dimension n x n. The elements of the matrix $W_{ij} = \omega_{ij}$ when $i \neq j$ and 0 otherwise.

The graph Laplacian matrix is a ***symmetric, positive semi-definite*** *matrix with precisely n real-valued, eigenvalues (if a matrix with real entries is symmetric then its eigenvalues are real) and n linearly independent non-zero eigenvectors.*

- **Symmetric** as $L^T = L$ and follows directly from the symmetry of D and W.

- **Positive semi-definite** A symmetric matrix is positive semi-definite if the quadratic form is at least zero. The quadratic form of the Laplacian graph $L \in \mathbb{R}^{nxn}$ is:

$$\begin{aligned}
&f'Lf = f'(D-W)f = f'Df - f'Wf \\
&\quad = \begin{bmatrix} f_1, \ldots, f_n \end{bmatrix} \begin{bmatrix} \sum\limits_{j=1}^{n} \omega_{1j} & & \\ & \ddots & \\ & & \sum\limits_{j=1}^{n} \omega_{nj} \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ \vdots \\ f_n \end{bmatrix} - \begin{bmatrix} f_1, \ldots, f_n \end{bmatrix} \begin{bmatrix} 0 & & & & \\ \omega_{2,1} & 0 & & & \\ \omega_{3,1} & \omega_{3,2} & 0 & & \\ \vdots & \vdots & & \ddots & \\ \omega_{n,1} & \omega_{n,2} & \ldots & & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ \vdots \\ f_n \end{bmatrix} \\
&\quad = f_1^2 \sum_{j=1}^{n} \omega_{1j} + f_2^2 \sum_{j=1}^{n} \omega_{2j} + \ldots + f_n^2 \sum_{j=1}^{n} \omega_{nj} - \Big(f_1 \sum_{j=1}^{n} f_j \omega_{j1} + f_2 \sum_{j=1}^{n} f_j \omega_{j2} + \ldots + f_n \sum_{j=1}^{n} f_j \omega_{jn} \Big) \\
&\quad = \sum_{i=1}^{n} f_i^2 \sum_{j=1}^{n} \omega_{ij} - \sum_{i=1}^{n} f_i \sum_{j=1}^{n} f_j \omega_{ji} = \frac{1}{2} \sum_{i,j=1}^{n} 2 f_i^2 \omega_{ij} - 2 f_i f_j \omega_{ji} = \frac{1}{2} \sum_{i,j=1}^{n} \omega_{ij} (f_i^2 + f_j^2 - 2 f_i f_j) \\
&\quad = \frac{1}{2} \sum_{i,j=1}^{n} \omega_{ij} (f_i - f_j)^2 \geq 0
\end{aligned} \tag{1}$$

  The Laplacian graph will be positive semi-definite for all vectors f, meaning that it will have non-negative eigenvalues. If f is an eigenvector of L, then we know that $f'Lf = f'\lambda f = f'f\lambda$, since $f'f$ and $f'Lf$ are non-negative values, then the eigenvalue λ must also be a non-negative value (this is not always true, it holds as the eigenvalues are real-valued).

  Said that, we can notice that the smallest eigenvalue will be 0. The number of eigenvectors will correspond to the number of components in the graph. For instance, if we have a fully connected graph then the eigenvector of the eigenvalue 0, will be the constant ones-vector $< 1, 1, ... >$, if we had k components we will have k non-trivial eigenvectors corresponding to the 0 eigenvalue, each eigenvector will indicate the connected nodes of the component, hence the eigenvectors will be indicator $\mathbb{1}$ vectors.

  If we take a look at (1), $f'Lf = \lambda = 0$ whenever $\omega_{ij}(f_i - f_j)^2 = 0 \; \forall i, j = 1, ..., n$. This condition will be satisfied if (i) $\omega_{ij} = 0$ and $f_i \neq f_j$, referring to self-connectivity cases or no connection between nodes, and (ii) the elements of the eigenvector $f_i = f_j > 0$, and $\omega_{ij} > 0$, which indicates that node i is connected to node j, and this is why the first (smallest) eigenvalue indicates the number of components of a graph. Furthermore k denotes the multiplicity of the eigenvector λ.

The unnormalized graph Laplacian matrix will look like:

$$\begin{bmatrix} \sum_{j=1}^{n}\omega_{1j} & & & & \\ -\omega_{21} & \sum_{j=1}^{n}\omega_{2j} & & & \\ -\omega_{31} & -\omega_{32} & \sum_{j=1}^{n}\omega_{3j} & & \\ \vdots & \vdots & \vdots & \ddots & \\ -\omega_{n1} & -\omega_{n2} & \dots & \sum_{j=1}^{n}\omega_{nj} & \end{bmatrix} L_{ij} = \begin{cases} D_{ii} & i = j \\ -\omega_{ij}, & i \neq j \end{cases}$$

Notice that each row/column will sum to zero.

Question: What does it mean by *unnormalized*?

If we re-scale the connectivity weights in our network, the eigenvectors will not be changed (the connectivity will remain the same), however the eigenvalues of the unnormalized Laplacian graph will be scaled by a constant. It is called *unnormalized* as the eigenvalue depends on ω_{ij}, we will need to fix a constant in order to normalize it.

Knowing that $f'Lf = f'f\lambda$. Eigenvectors can be normalize by requiring that $f'f = 1$, i.e we divide each element of the eigenvector f by $\sqrt{f_1^2 + f_2^2 + ... + f_n^2}$. By this procedure we can see that the eigenvector f does not depends on the weights, while the eigenvalue does, as it can be seen in (1), if we normalize the eigenvector f, $f'Lf = \lambda$, clearly λ depends on the value of ω_{ij}.

1.4 Normalized graph Laplacian

1.4.1 Symmetric Normalized graph Laplacian

$$\begin{aligned} L_{sym} &= D^{-1/2}LD^{-1/2} = D^{-1/2}(D - W)D^{-1/2} \\ &= D^{-1/2}DD^{-1/2} - D^{-1/2}WD^{-1/2} = I - \frac{1}{\sqrt{D}}W\frac{1}{\sqrt{D}} \end{aligned} \tag{2}$$

Where I denotes the identity matrix, L, the unnormalized Laplacian graph and W the adjacency matrix. The normalized symmetric Laplacian graph is an n x n positive semi-definite matrix. This property can be proved by computing the quadratic form such that $f'L_{sym}f \geq 0$ for all $f \in \mathbb{R}^n$

$$
\begin{aligned}
f' L_{sym} f &= f' I f - f' \frac{1}{\sqrt{D}} W \frac{1}{\sqrt{D}} f \\
&= \left[f_1, \ldots, f_n\right] \begin{bmatrix} f_1 \\ \vdots \\ \vdots \\ f_n \end{bmatrix} - \left[\tfrac{f_1}{\sqrt{D_{11}}}, \ldots, \tfrac{f_n}{\sqrt{D_{nn}}}\right] \begin{bmatrix} 0 & & & & \\ \omega_{2,1} & 0 & & & \\ \omega_{3,1} & \omega_{3,2} & 0 & & \\ \vdots & \vdots & & \ddots & \\ \omega_{n,1} & \omega_{n,2} & \ldots & & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{D_{11}}} & & & \\ & \frac{1}{\sqrt{D_{22}}} & & \\ & & \ddots & \\ & & & \frac{1}{\sqrt{D_{nn}}} \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ \vdots \\ f_n \end{bmatrix} \\
&= \sum_{i=1}^{n} f_i^2 - \sum_{i=1}^{n} \frac{f_i}{\sqrt{D_{ii}}} \Big(\sum_{j=1}^{n} \frac{f_j}{\sqrt{D_{jj}}} \omega_{ji} \Big) = \sum_{i=1}^{n} f_i^2 \frac{\sum_{j=1}^{n} \omega_{ij}}{D_{ii}} - \sum_{i,j=1}^{n} f_i f_j \Big(\frac{\omega_{ij}}{\sqrt{D_{ii}} \sqrt{D_{jj}}} \Big) \\
&= \sum_{i,j=1}^{n} f_i^2 \frac{\omega_{ij}}{D_{ii}} - f_i f_j \Big(\frac{\omega_{ij}}{\sqrt{D_{ii}} \sqrt{D_{jj}}} \Big) = \frac{1}{2} \Big(\sum_{i,j=1}^{n} \omega_{ij} \Big(\frac{f_i^2}{D_{ii}} + \frac{f_j^2}{D_{jj}} - 2 \frac{f_i f_j}{\sqrt{D_{ii}} \sqrt{D_{jj}}} \Big) \Big) \\
&= \frac{1}{2} \sum_{i,j=1}^{n} \omega_{ij} \Big(\frac{f_i}{\sqrt{D_{ii}}} - \frac{f_j}{\sqrt{D_{jj}}} \Big)^2 \geq 0
\end{aligned}
\tag{3}
$$

The Normalized graph Laplacian will look like:

$$
\begin{bmatrix} 1 - \omega_{11}/D_{11} & & & & \\ -\omega_{21}/\sqrt{D_{22}}\sqrt{D_{11}} & 1 - \omega_{22}/D_{22} & & & \\ -\omega_{31}/\sqrt{D_{33}}\sqrt{D_{11}} & -\omega_{32}/D_{33}\sqrt{D_{22}} & 1 - \omega_{33}/D_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ -\omega_{n1}/\sqrt{D_{nn}}\sqrt{D_{11}} & -\omega_{n2}/\sqrt{D_{nn}}\sqrt{D_{22}} & \ldots & & 1 - \omega_{nn}/D_{nn} \end{bmatrix} L_{sym}^{ij} = \begin{cases} 1 & i = j \\ -\omega_{ij}/\sqrt{D_{ii}}\sqrt{D_{jj}}, & i \neq j \end{cases}
$$

Question: What does it mean by *normalized*?

If we now re-scale ω_{ij} by a constant factor, it will not have any effect on the Normalized graph Laplacian eigenvalues, given that the terms of the quadratic form are proportional to the elements of the diagonalized matrix (which depend on the weights). The constant factor multiplying ω_{ij} in (3) will cancel out with the scaling factor dividing the quadratic form terms.

1.4.2 Random walk normalized graph Laplacian

A discrete time random walk on a graph can be seen as a Markov chain where each node (data point) denotes an element of the state space. The Random walk normalized Laplacian matrix is a positive semi-definite non-symmetric matrix.

$$
L_{rw} = D^{-1} L = D^{-1}(D - W) = I - D^{-1} W = I - P \tag{4}
$$

The matrix $D^{-1}W$ is equivalent to the n x n transition matrix $P \in \mathbb{R}^{nxn}$ with elements given by $P_{ij} = P(j|i) = \frac{\omega_{ij}}{D_{ii}}$ and represents the conditional probability of transitioning to node j from node i in a time step.

The transition probability matrix is generally not symmetric, and this is why the Random walk normalized Laplacian matrix is not symmetric.

$$D^{-1}W = P = \begin{bmatrix} \omega_{11}/D_{11} & & & & \\ \omega_{21}/D_{22} & \omega_{22}/D_{22} & & & \\ \omega_{31}/D_{33} & \omega_{32}/_{33} & \omega_{33}/D_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \omega_{n1}/D_{nn} & \omega_{n2}/_{nn} & \dots & & \omega_{nn}/D_{nn} \end{bmatrix}, P_{ij} = \begin{cases} 0 & i = j \\ \omega_{ij}/D_{ii}, & i \neq j \end{cases}$$

P is a row stochastic matrix, meaning that $\sum_{j=1}^{n} w_{ij}/D_{ii} = 1$. The eigenspectrum of L_{rw} and P will differ by a constant factor of 1 while their eigenvectors will be the same.

> λ is is an eigenvalue of L_{rw} with eigenvector v if and only if λ is an eigenvalue of L_{sym} with eigenvector $w = D^{1/2}v$

$$\begin{aligned} L_{sym}\ w &= \lambda w \\ (I - D^{-1/2}WD^{-1/2})D^{1/2}v &= \lambda D^{1/2}v \\ (D^{1/2} - D^{-1/2}W)v &= \lambda D^{1/2}v \\ (I - \lambda)D^{1/2}v &= D^{-1/2}Wv \\ (I - \lambda)v &= D^{-1}Wv \\ (I - D^{-1}W)v &= \lambda v \\ L_{rw}\ v &= \lambda v \end{aligned} \tag{5}$$

From the above we can also notice that the eigenvector v with the largest eigenvalue for P is the eigenvector v with the smallest eigenvalue λ for L_{rw}.

Generally it would be more convenient to work with a symmetric matrix such as L_{sym}, as it has nicer properties compared to L_{rw}. However, L_{sym} and L_{rw} offer two perspectives to the same problem.

> **Question:** Why do we set $\omega_{ii} = 0$ as a condition?

The question is, why do we set the self-connectivity to zero and not any other number?. By convention we assign a value of zero to self-connectivity cases because when computing the weights we are only interested in knowing the pairwise similarity between points and not with itself.

If we were to assign a non-zero weight to self-connectivity cases we would be assigning a high connectivity weight to a case that does not contribute with relevant information in our similarity graph.

In terms of

- **The unnormalized Laplacian graph**, if we assign self-connectivity weights, it will not have any effect on the matrix diagonal as it will cancel out when subtracting the diagonal and adjacency matrix, meaning that the elements in the unnormalized Laplacian matrix will not change. In terms of the eigenspectrum, neither the eigenvectors nor eigenvalues will change.

- **Random walk normalized graph Laplacian** if we assign self-connectivity weights, the matrix elements will be changed. The transition probability matrix will be modified as we will now have non-zero self-transitioning probabilities, i.e the terms in the matrix $p_{ij} = w_{ij}/D_{ii} > 0$, for $i = j$.

A side effect of assigning self-connectivity weights is that these weights will be higher than the rest of connectivity weights, meaning that the probability of transitioning to a different state will be lower. Under this scenario, both the eigenvalue and eigenvectors will change.

- **Symmetric normalized graph Laplacian** From (5) we know that the eigenvalues of L_{rw} and L_{sym} are the same if and only if their eigenvectors are related by a constant, i.e $w = D^{1/2}v$. Hence, if we include self-connectivity weights both normalized Laplacian matrices will be modified as well as their eigenvalues and eigenvectors.

There is a close relationship between the eigenspectrum of L_{sym} and L_{rw}, but there is not such a relationship between the eigenspectrum of the unnormalized graph Laplacian L and the normalized graph Laplacian L_{sym} and L_{rw}. This non-relationship also follows from the fact that the unnormalized graph Laplacian is invariant to self-transitioning weights, while the normalized Laplacian graphs are not invariant.

2 Meeting Summary 07/10/2020

We started by discussing about further reasons of why we set self-connectivity weights $\omega_{ij} = 0$ for $i = j$. Notice that as the unnormalized Laplacian graph is invariant to any transformation, and so setting self-connectivity weights to zero will only have an effect on the normalized Laplacian matrices, some further reasons can be:

- Setting the self-connectivity weights to zero can be computationally advantageous as it will make computations faster.
- Setting non-zero self-connectivity weights prevent exploring the manifold: For instance, if we have an isolated point with no neighbors in the radius defined by a Gaussian Kernel, the connective-weights to the rest of the nodes in the network will be very weak compared to the self-transition weight, which in terms of the self-transition conditional probability will dominate over the rest of transitioning probabilities i.e $p_{ij} \sim 1$ for $i = j$.

  The problem of this scenario is that although the probability of transitioning to this isolated node, could be very small or not, if we do transition, there is a high probability of staying trapped in the same isolated node for many time steps. Hence, if we want to study all possible paths of traveling from one node to another we will have to take into account the time the node will travel to itself before it jumps out, which will make the problem more complicated.

2.0.1 The commute time distance CTD

We can write it in terms of both, the eigenspectrum and eigenvalue of either the unnormalized and normalized Laplacian graph.

> **Question:** Does the CTD c_{ij} will depends on non-zero self-connectivity weights?

We can write c_{ij} in terms of the eigenspectrum of the unnormalized and normalized Laplacian graph. This fact could lead to confusion as we know that if we set non-zero self-transitioning weights, the eigenspectrum of the unnormalized Laplacian graph will remain invariant while in the case of the eigenspectrum of the normalized Laplacian graph will not. Hence, it is not trivial to see that c_{ij} will also remain invariant to non-zero self-transitioning weights as it can be expressed in terms of the eigenspectrum of L_{sym} which is not invariant.

Notice that this will not be the case for other quantities computed by using the normalized graph Laplacian. This in-variance property will affect only the CTD.

- In terms of the **unnormalized graph Laplacian:**

$$c_{ij} = \mathrm{vol}(V) \sum_{k=2}^{n} \frac{1}{\lambda_k} \Big(v_{kj} - v_{ki} \Big)^2$$

- In terms of the **normalized graph Laplacian:**

$$
\begin{aligned}
c_{ij} &= H(i,j) + H(j,i) \\
&= \mathrm{vol}(V)\sum_{k=2}^{n}\frac{1}{\lambda_k}\Big(\frac{v_{kj}^2}{d_j} - \frac{v_{ki}v_{kj}}{\sqrt{d_i D_{jj}}}\Big) + \mathrm{vol}(V)\sum_{k=2}^{n}\frac{1}{\lambda_k}\Big(\frac{v_{ki}^2}{D_{ii}} - \frac{v_{kj}v_{ki}}{\sqrt{D_{jj}D_{ii}}}\Big) \\
&= \mathrm{vol}(V)\sum_{k=2}^{n}\frac{1}{\lambda_k}\Big(\frac{v_{kj}^2}{D_{jj}} + \frac{v_{ki}^2}{D_{ii}} - 2\frac{v_{ki}v_{kj}}{\sqrt{D_{ii}D_{jj}}}\Big) \\
&= \mathrm{vol}(V)\sum_{k=2}^{n}\frac{1}{\lambda_k}\Big(\frac{v_{kj}}{\sqrt{D_{jj}}} - \frac{v_{ki}}{\sqrt{D_{ii}}}\Big)^2
\end{aligned}
\tag{6}
$$

From this formulation we cannot see that c_{ij} is not invariant to self-connectivity weights so given that:

$$
c_{ij} = \sum_{k=2}^{n}\frac{1}{\lambda_k^{un}}\Big(v_{kj}^{un} - v_{ki}^{un}\Big)^2 = \sum_{k=2}^{n}\frac{1}{\lambda_k^{norm}}\Big(\frac{v_{kj}^{norm}}{\sqrt{D_{jj}}} - \frac{v_{ki}^{norm}}{\sqrt{D_{ii}}}\Big)^2
$$

We know that c_{ij} is invariant to self-connectivity weights as it can be written in terms of L (unnormalized graph Laplacian), which will mean that the terms on the RHS of the equation above (although computed with non-invariant components - referring the the eigenvalues, eigenvectors and diagonal matrix components) will result in an invariant term.