Over-smoothing and diffusion dynamics on graphs

Master Thesis

University: Imperial College London
Departement: Mathematics
Supervisor: Jeroen Lamb
Date: September 2023
Student name: Jean Adrien Lagesse
Student CID: 02331224

Abstract

Over-smoothing is a recurrent problem when working with Graph Neural Networks that severely limits the expressiveness of well-known architectures. In this report we have gathered from different papers a mathematically tractable definition of this problem, we proposed a proof of the exponential over-smoothing of the isotropic diffusion equation on graphs and generalized it to anisotropic positive diffusion dynamics. To prove these theorems, we introduced different pseudo-Euclidean spaces adapted to measure over-smoothing in different use cases. Finally, we implemented a fast GPU-optimized algorithm based on the Graph Fourier transformation to analyze in practice this phenomenon for Erdos-Rényi random graphs.

Integrity statement on plagiarism

The work contained in this thesis is my own work unless otherwise stated.

Acknowledgment

I would like to thank Professor Jeroen Lamb and Victoria Klein for their help and guidance.

Notations

$\mathbf{G} = (\mathbf{V}, \mathbf{E})$: A simple connected undirected graph with vertices set $\{1, ..., V\}$ and edges in $\{1, ..., V\}^2$.

$\mathcal{X}^{\mathbf{l}}(\mathbf{G})$: The set of all l-dimensional signals on the vertices of the graph G, associating every vertex with a vector in $\mathbb{R}^l$.

$\mathcal{H}^{\mathbf{l}}(\mathbf{G})$: The set of all l-dimensional signals on the edges of the graph G, associating every edge with a vector in $\mathbb{R}^l$. The signals on G are alternating: if $\epsilon \in \mathcal{H}^l(G)$ and $(i, j) \in E$,then $\epsilon(i, j) = -\epsilon(j, i)$.

$\mathbf{i} \sim \mathbf{j}$: The vertices i and j are neighbours in G (*i.e.* $(i, j) \in E$).

$\bigoplus$: Aggregation operator.

$\mathbf{A}$: Adjacency matrix of the graph G.

$\mathbf{\Delta}$: Laplacian of the graph G.

$\bar{\mathbf{\Delta}}$: Normalized Laplacian of the graph G.

$\tilde{\mathbf{\Delta}}$: Normalized augmented Laplacian of the graph G.

$\mathbf{E_{\Delta}}(\mathbf{X})$: Dirichlet Energy associated with the Laplacian Δ of a signal $X \in \mathcal{X}^l(G)$ on the graph G.

$\mathbf{E_{\bar{\Delta}}}(\mathbf{X})$: Dirichlet Energy associated with the normalized Laplacian Δ of a signal $X \in \mathcal{X}^l(G)$

$\mathbf{E_{\tilde{\Delta}}}(\mathbf{X})$: Dirichlet Energy associated with the augmented normalized Laplacian Δ of a signal $X \in \mathcal{X}^l(G)$ on the graph G.

$\mathbf{E_S}(\mathbf{X})$: Dirichlet Energy associated with the symmetric positive semi-definite matrix S of a signal $X \in \mathcal{X}^l(G)$ on the graph G.

$< \mathbf{X}, \mathbf{Y} > = tr(X^T Y)$: inner product on $\mathcal{X}^l(G)$

$||\mathbf{X}||_{\mathbf{2}}$: L^2 vector norm.

$||\mathbf{M}||_{\mathbf{2}}$: Matrix norm associated with the L^2 vector norm. $||M||_2 = \max_{||X||_2=1} ||MX||_2$

$< \mathbf{X}, \mathbf{Y} >_{\mathbf{S}} = tr(X^T SY)$: pseudo inner product induced by a symmetric, positive, semi-definite matrix S.

$||\mathbf{M}||_{\mathbf{S}}$: Matrix norm associated with the Dirichlet energy. $||M||_S = \sup_{E_S(X)=1} E_S(MX)$

$\widehat{\mathbf{X}}$: Graph Fourier Transform of a signal $X \in \mathcal{X}^l(G)$

$\widehat{\mathbf{M}}$: Graph Fourier Transform of the matrix M.

$\lambda_{\mathbf{1}} \leq ... \leq \lambda_{\mathbf{V}}$: Eigenvalues of the symmetric matrix S order in ascending order. If which matrix S is not clear, we write $\lambda_i(S)$.

$\lambda_{\mathbf{min}}, \lambda_{\mathbf{max}}$: Smallest and largest eigenvalue of a symmetric matrix.

$\dot{\mathbf{X}}(\mathbf{t})$: time derivative of X

Contents

1 Introduction

With the fast rise of Machine Learning finding new architectures that work for new data types has been a priority. The classical data types have always been Euclidean: text data can be seen as a 1-dimensional, images are 2-dimensional, etc... Yet, for many problems, Euclidean data types are not very well suited. Graphs being a very well-known and powerful data structure in computer science it was very natural to use it in several problems. The paper (Scarselli et al. [9]) presented the Graph Neural Network architecture. Since then, improving the original Graph Neural Network architecture has been a problem that many researchers are working on.

In the past few years, hundreds of different architectures for Graph Neural Networks have been presented, most of them can be classified into three categories:

1. Graph Convolution Network: These architectures can be seen as a generalization of the Convolution Neural Networks used mainly in Computer Vision to Graphs. They are based on an approximation of convolutions on graphs (Kipf et al. [7]).

2. Graph Attention Networks: These architectures can be seen as a generalization of the transformers used mainly in Natural Language Processing to graphs (Veličković et al. [11]).

3. Message Passing Neural Networks: A strictly more powerful architecture than the Graph Attention Network (Gilmer et al. [5]).

The evolution of Graph Neural Networks closely follows the rest of the Machine Learning field. Several ideas first implemented in Computer Vision and Natural Language processing have been implemented in Graph Neural Networks. In this report, we are particularly interested in the Residual Connection Networks paper (He et al. [6]) that was later used to define the Neural Ordinary Differential Equation Network architecture (Chen et al. [3]) and enables the use of continuous-time processes as Neural Networks. These two papers enable us to consider the discrete layers of a Graph Neural Network as a continuous function which simplifies the study of the dynamics on the graph.

2 Graph Neural Network

2.1 Graphs and signals on graphs

Let's consider a graph $G = (V, E)$ where the vertices are numbered from 1 to V and where the edges in E are the pairs (i, j) for some $(i, j) \in \{1, ..., V\}^2$, moreover we consider that the graph is simple, undirected and connected.

On this graph G, we will assign features to each vertex, those features are vectors of a fixed size l, hence, we can represent all the features of the graph as a $V \times l$ matrix that we will call X. We say that X is an l-dimensional signal on G and we call $\mathcal{X}^l(G)$ the set of all l-dimensional signals.

Similarly, we can assign features to the edges of G. Let $\epsilon : \{1, ..., V\}^2 \longrightarrow \mathbb{R}^l$ such that for all $(i, j) \in \{1, ..., V\}^2$, $\epsilon(i, j) = -\epsilon(j, i)$, and such that if $(i, j) \notin E$ then $\epsilon(i, j) = 0$. We say that ϵ is an l-dimensional edge signal on G and we call $\mathcal{H}^l(G)$ the set of all l-dimensional edge signals.

2.2 Architecture

In the most general definition, a graph neural network is a function H_θ parameterized by θ, that takes as input a graph G with an l_{in}-dimensional signal X_{in} and outputs an l_{out}-dimensional signal. Moreover, the function H_θ should be able to take any graph G as its input but we will require that l_{in} and l_{out} (the number of features we have per vertices) stay the same.

A graph neural network is a type of Neural Network that takes as an input a graph and gives back another graph. A key element of Graph Neural Networks is that the same network can be used on graphs of different structures and different sizes, whereas classical neural networks have a fixed vector size as an input. This is very useful when working on heterogeneous data such as molecules: each molecule has a different number of atoms and covalent bonds, making it very hard to have a fixed vector size representation.

2.3 Optimization and training of a Graph Neural Network

Let's consider that we have a training set $\mathbb{X} = (G_i, X_i)_{i \in \{1,...,N\}}$ consisting of N graphs and N l_{in}-dimensional signals. Moreover, we have $\mathbb{Y} = (G_i, Y_i)_{i \in \{1,...,N\}}$ consisting

of N graphs and N l_{out}-dimensional signals. Our aim is to find the optimal θ^* such that H_{θ^*} predict for each $i \in \{1, ..., V\}$ Y_i from X_i. In practice, we try to solve:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \mathcal{L}(H_\theta(X_i, G_i), Y_i) + \mathcal{L}_{reg}(\theta) \tag{1}$$

where $\mathcal{L}$ is a loss function and $\mathcal{L}_{reg}$ is a regularization penalty.

When the function H_θ is differentiable, the usual choice to find an approximate solution of 1 is to use the stochastic gradient descent algorithm.

2.4 Aggregation function

The most important requirement is that our function H work on a graph of any structure *i.e.* with a different number of vertices and different edges, moreover we want to use this structure as it carries information about the data on which we are working. Intuitively, we want to learn the output feature of a vertex by analyzing its neighbourhood in the graph; to be able to extract this information, we need an *aggregation function*, this function will take an arbitrary number of neighbours of a vertex and aggregate it into a fixed size vector, in the rest of the report, this aggregation function will be denoted as $\bigoplus$, and will aggregate any number of feature vector $\{v_1, ..., v_m\}$ into another vector $\bigoplus_{i=1}^m v_i$. To make sense, the *aggregation function* must be invariant under permutation because there is no canonical order of vertices in a graph, hence for a permutation σ, we must have that $\bigoplus_{i=1}^m v_i = \bigoplus_{i=1}^m v_{\sigma(i)}$. This function will compute a *representation* of the neighbourhood of a vertex. Let $\{v_1, ..., v_m\}$ be vectors in $\mathbb{R}^l$, here are a few examples of *aggregation functions*:

- $\bigoplus_{i=1}^m v_i = \dfrac{1}{m} \sum_{i=1}^m v_i$
- $\bigoplus_{i=1}^m v_i = (max(v_1[1], ..., v_m[1]), ..., max(v_1[l], ..., v_m[l])^T$
- $\bigoplus_{i=1}^m v_i = (min(v_1[1], ..., v_m[1]), ..., min(v_1[l], ..., v_m[l])^T$

2.5 Building a Graph Neural Network by stacking layers

Now that we have defined the aggregation function, we can give a general formula for the function H. Let's consider $l_{input} = l_0, l_1, ..., l_N = l_{output}$ the dimension of the input signal, output signal and intermediate signals that we will use. We will also break down the function H in several layers such that $H = h_0 \circ ... \circ h_{N-1}$, where each

layer h_i will take as an input the graph G and it's l_i-dimensional signal $X_i \in \mathcal{X}_i^l(G)$ and output a new l_{i+1}-dimensional signal $X_{i+1} \in \mathcal{X}_{i+1}^l(G)$

Now we will give the general formula for each layer, let's consider the layer h_k that takes as an input X_k and outputs X_{k+1}, we will denote $x_j^{(k)}$ the feature vector associated with the j^{th} vertices in X_k. The function h_k is composed of two parameterized functions $\gamma^{(k)}$ and $\phi^{(k)}$ (usually *multi-layer perceptrons*) and we can compute the new signal X_{k+1} with the following formula:

$$x_i^{(k+1)} = \gamma^{(k)}(x_i^{(k)}, \bigoplus_{j \sim i} \phi^{(k)}(x_i^{(k)}, x_j^{(k)}))$$

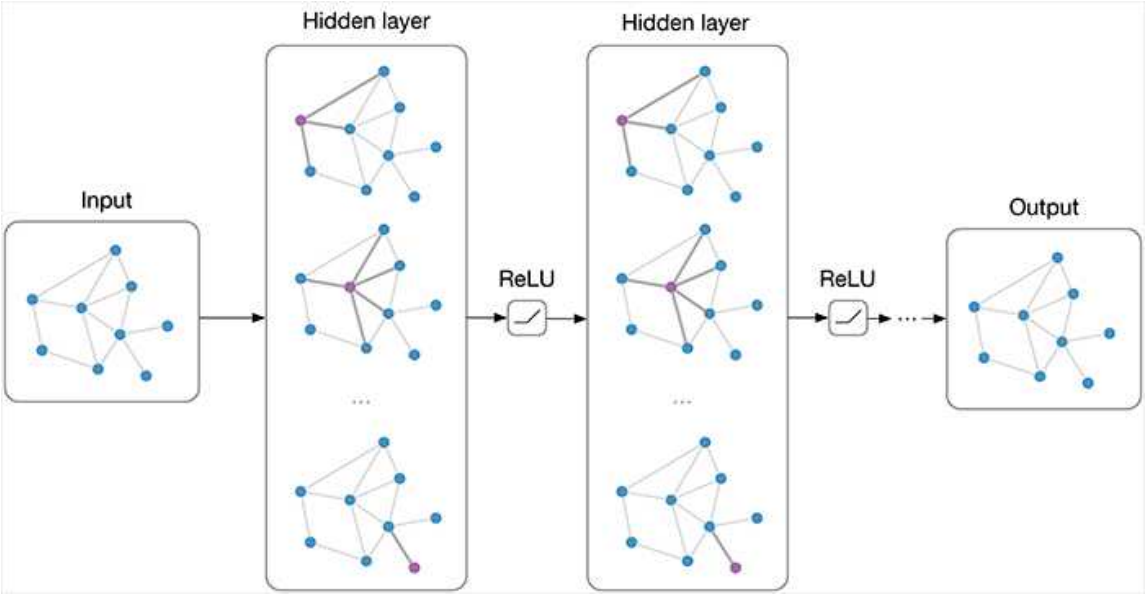

Figure 1: Visualization of the layers of a Graph Neural Network: We can see a graph G with a feature matrix of dimension $V \times l_i$ as l_i graph with a feature matrix of dimension $V \times 1$, each of these graph represents a different learned information. (source: Thomas Kipf - Deep learning with graph-structured representations)

2.6 Classification of Graph Neural Network architectures

In the literature, many different Graph Neural Networks architectures have been presented, however most of them can be classified in the following way:

Definition 1 (The expressive power of Graph Neural Networks). .

- **Convolution Graph Neural Networks**: this architecture is the simplest and is described by the following equation for each layer:

$$x_i^{(k+1)} = \gamma^{(k)}(x_i^{(k)}, \bigoplus_{j \sim i} H_{i,j} x_j^{(k)})$$

  N.B: H a matrix.

- **Graph Attention Networks**: this architecture is in between the Convolution Graph Neural Network and the Message Passing Neural Network, rather than simply aggregating the features, we will perform a weighted aggregation:

$$x_i^{(k+1)} = \gamma^{(k)}(x_i^{(k)}, \bigoplus_{j \sim i} a^{(k)}(x_i^{(k)}, x_j^{(k)}) x_j^{(k)})$$

  N.B: The function $a^{(k)}$ compute a weight in $\mathbb{R}$.

- **Message Passing Neural Networks**: this architecture is the more general and the more expressive, for each pair of features we will compute a completely new feature to be aggregated. It is the same equation as in the previous section:

$$x_i^{(k+1)} = \gamma^{(k)}(x_i^{(k)}, \bigoplus_{j \sim i} \phi^{(k)}(x_i^{(k)}, x_j^{(k)}))$$

Each one of those architectures or more powerful than the previous one, however, this complexity comes at a cost. Indeed, as the neural network gets more expressive, it is harder to train. Nevertheless, depending on the characteristics of the data on which we are working, using a more expressive model can be required.

2.7 From discrete layers to continuous layers

Residual Networks (He et al. [6]) were first introduced in Computer Vision to have a better propagation of the gradient in the Neural Network (useful for the stochastic descent algorithm). Applying this to Graph Neural Networks means to parameterize differently the layer-wise equation:

$$x_i^{(k+1)} = x_i^{(k)} + \gamma^{(k)}(x_i^{(k)}, \bigoplus_{j \sim i} \phi^{(k)}(x_i^{(k)}, x_j^{(k)}))$$

In *Neural Ordinary Differential Equations* (Chen et al. [3]), it was suggested that for Residual Networks, it is possible to consider that the Neural Network layers are continuous. Applying this to the Graph Neural Network architecture yields to the following differential equation:

$$\dot{x}_i(t) = \gamma(t)(x_i(t), \bigoplus_{j \sim i} \phi(t)((x_i(t), x_j(t))) \tag{2}$$

With this formalism $t \mapsto \gamma(t)$ and $t \mapsto \phi(t)$ assigns to each t a parameterized function. Ways to build and optimize those function are presented in (Chen et al. [3]) but are not relevant to the rest of this report.

The equation 2 is linked to the Message Passing Neural Network update equation by the Euler discretization scheme, for $\tau = 1$ we have:

$$\frac{x_i(t+\tau) - x_i(t)}{\tau} = \gamma(t)(x_i(t), \bigoplus_{j \sim i} \phi(t)((x_i(t), x_j(t)))$$

$$x_i(t+\tau) = x_i(t) + \tau\gamma(t)(x_i(t), \bigoplus_{j \sim i} \phi(t)((x_i(t), x_j(t)))$$

$$x_i(t+1) = x_i(t) + \gamma(t)(x_i(t), \bigoplus_{j \sim i} \phi(t)((x_i(t), x_j(t)))$$

Hence, we can see that discrete and continuous layers follow the same dynamics when the number of layers is big (*i.e.* when τ is small).

3 Properties and limitations of Graph Neural Networks

3.1 Importance of deep Graph Neural Networks

Deep Learning is a sub-field of Machine Learning where several layers are stacked one after each other, using deep learning enables learning very complex relationships in the data and has resulted in state-of-the-art methods in several domains such as Computer Vision and NLP. Empirically, it has been shown that large and deep models perform better than shallow models. Large Language Models consist of tens of layers (Touvron et al. [10]) and image recognition architectures can rise to hundreds of layers (He et al. [6]).

In Graph Neural Networks, having many layers has another important role:

Theorem 1. — *Consider a Graph Neural Network with N layers, the output feature of a vertex v depends exactly on the features of all the vertices at a distance of N or less to v.*

Hence, to be able to learn interactions between long-distance vertices, it is necessary to have very deep Graph Neural Networks. However, several problems arise when considering deep GNN architectures. We will investigate those problems by considering the dynamics on the graph in the following parts of the report.

3.2 Limitations of deep Graph Neural Networks

Definition 2. Graphs can be classified in two main categories:

- **Homophilic graphs**: We call a graph homophilic if we expect that neighboring vertices will share the same features. An example of such graphs is social media graphs: if two persons are friends on a social network (*i.e* are neighbors in the friendship graph) we can expect that they will share the same features such as location, political views, etc...
- **Heterophilic graphs**: We call a graph heterophilic if we expect that neighboring vertices don't share the same features. An example of such graphs is the graph representation of a molecule (*i.e* the graph where vertices are atoms and edges represent the covalent bonds), indeed, in molecules, very different atoms share covalent bonds hence they will have different features.

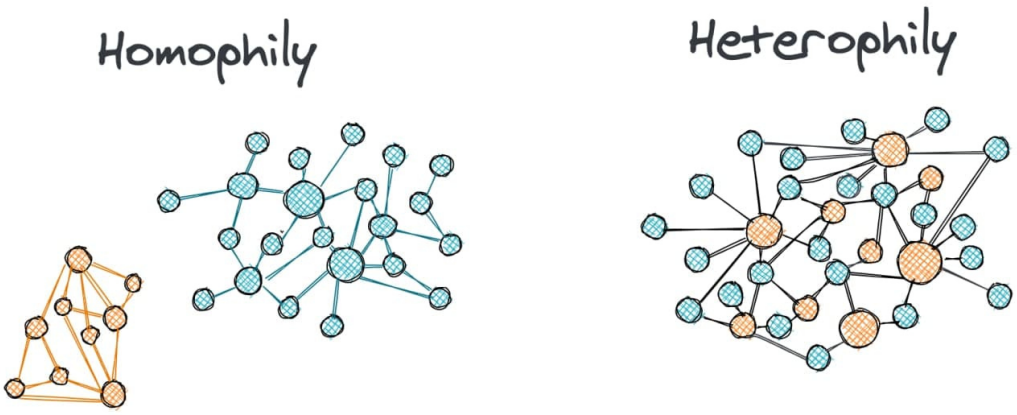

Figure 2: Visualization of heterophilic and homophilic graphs.
(source: https://graphml.substack.com/p/gml-newsletter-homophily-heterophily)

On homophilic graphs simple models such as a Graph Convolutional Network can yield very good results, however, on heterophilic graphs, the neural network cannot distinguish vertices that are similar to one another from the ones that should share very different features, hence, during the aggregation step all this information is lost and this yields to an output graph that is very smooth (neighboring nodes are very similar): that problem is commonly known as **over-smoothing**.

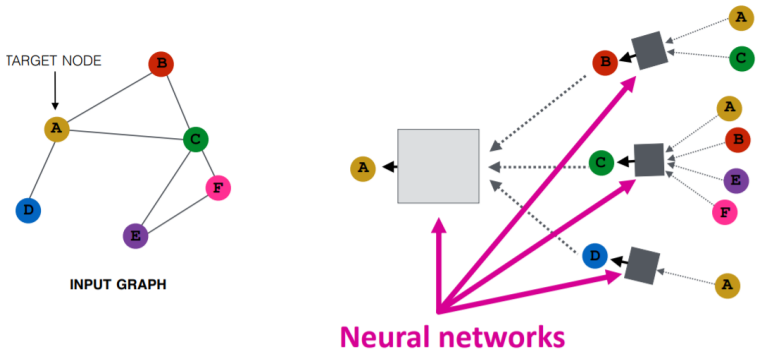

Figure 3: Unwinding of the layers of a Graph Neural Network from the point of view of a single vertex. (source: https://medium.com/neuralspace/graphs-neural-networks-in-nlp-dc475eb089de)

Over-smoothing can also arise from very deep Graph Neural Networks, as seen in 3,

the vertices present when unwinding the Graph Neural Network are often repeated, hence when we have many layers, the information learned about a vertex is nearly the same as the its neighbours, resulting in over-smoothing.

In addition to over-smoothing, another well know problem when using Graph Neural Network is **bottlenecks**. As we can see in 3, with only a few layers and a few neighbouring vertices, the information that a vertices contains is very packed, which means that it is very hard for the neural network to use all the available information, this can become quite a problem when there are very few edges connecting different dense part of the graph, just as shown in 4.

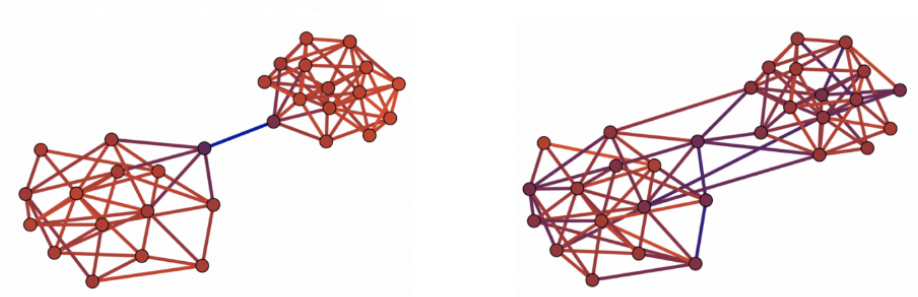

Figure 4: On the left we can see that a bottleneck will appear, information will difficultly flow between the right and left part of the graph. On the graph on the right, because there is more edges connecting the two parts, the information will be able to flow. (source: https://blog.twitter.com/engineering)

4 Spectral Graph Theory and Dirichlet Energy

To study Graph Convolution Networks and to prove that they are subject to over-smoothing, we will need the formalism of Graph Spectral Theory, a subset of graph theory that studies the relationships between the eigenvalues of the graph Laplacian and properties of the graph. Most of the definitions are inspired by (Chung [4]), however, based on the formula we used for the graph Laplacian, different properties emerge.

In addition, we define a new pseudo-Euclidean space that is adapted to study the Dirichlet energy of a graph. As we will see, the Dirichlet energy is linked to the Laplacian and we generalize the definition of the Dirichlet Energy to other types of graph Laplacians.

4.1 Common definitions

Let $G = (V, E)$ be a simple connected undirected graph.

Definition 3 (Adjacency matrix). The adjacency matrix of G is a matrix $A \in \mathcal{M}_{V\times V}(\{0,1\})$ such that for $i, j \in \{1, ..., V\}$, $A_{i,j} = 1$ if and only is $(i, j) \in E$ and $A_{i,j} = 0$ otherwise.

The augmented adjacency matrix of G is $\tilde{A} = A + \mathbb{I}d \in \mathcal{M}_{V\times V}(\{0,1\})$. It is the adjacency matrix of G augmented with self-loops.

Definition 4 (Degree matrix). The degree matrix of G is the diagonal matrix $D = diag(d_1, ..., d_V) \in \mathcal{M}_{V\times V}(\mathbb{N})$.

The augmented degree matrix of G is the diagonal matrix $\tilde{D} = diag(d_1 + 1, ..., d_V + 1) = D + \mathbb{I}d \in \mathcal{M}_{V\times V}(\mathbb{N})$. It is the degree matrix of G augmented with self-loops.

Definition 5 (Graph Laplacian). The graph Laplacian of G is the matrix $\Delta = D - A$.

The normalized graph Laplacian of G is the matrix $\bar{\Delta} = D^{-\frac{1}{2}}(D - A)D^{\frac{1}{2}}$

The augmented normalized graph Laplacian of G is the matrix

$$\begin{aligned}\tilde{\Delta} &= \tilde{D}^{-\frac{1}{2}}\Delta\tilde{D}^{-\frac{1}{2}} \\ &= \tilde{D}^{-\frac{1}{2}}(\tilde{D}-\tilde{A})\tilde{D}^{-\frac{1}{2}} \\ &= \mathbb{I}d - \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\end{aligned}$$

To have a a better intuition of how these operators act on a signal we can consider a 1-dimensional signal $X \in \mathbb{R}^V$on the graph G and let $i \in \{1, ..., V\}$ be a vertex of G:

$$\begin{aligned}\Delta(X)_i &= (DX)_i - (AX)_i \\ &= d_i x_i - \sum_{j\sim i} x_j \\ &= \sum_{j\sim i} x_i - x_j\end{aligned}$$

From that formula we can deduce:

$$\begin{aligned}\bar{\Delta}(X) &= \bar{D}^{-\frac{1}{2}}\Delta\bar{D}^{-\frac{1}{2}}X \\ &= \bar{D}^{-\frac{1}{2}}\Delta(\frac{x_i}{\sqrt{d_i}})_{i\in\{1,...,V\}} \\ &= \bar{D}^{-\frac{1}{2}}(\sum_{j\sim i}\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}})_{i\in\{1,...,V\}} \\ &= (\frac{1}{\sqrt{d_i}}\sum_{j\sim i}\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}})_{i\in\{1,...,V\}}\end{aligned}$$

and also:

$$\begin{aligned}\tilde{\Delta}(X) &= \tilde{D}^{-\frac{1}{2}}\Delta\tilde{D}^{-\frac{1}{2}}X \\ &= \tilde{D}^{-\frac{1}{2}}\Delta(\frac{x_i}{\sqrt{d_i+1}})_{i\in\{1,...,V\}} \\ &= \tilde{D}^{-\frac{1}{2}}(\sum_{j\sim i}\frac{x_i}{\sqrt{d_i+1}} - \frac{x_j}{\sqrt{d_j+1}})_{i\in\{1,...,V\}} \\ &= (\frac{1}{\sqrt{d_i+1}}\sum_{j\sim i}\frac{x_i}{\sqrt{d_i+1}} - \frac{x_j}{\sqrt{d_j+1}})_{i\in\{1,...,V\}}\end{aligned}$$

4.2 Properties of the Laplacian

Theorem 2. — *Δ, $\bar{\Delta}$ and $\tilde{\Delta}$ are symmetric, positive semi-definite matrices.*

PROOF : The matrix D is diagonal hence it is symmetric, moreover, because G is undirected, if $(i,j) \in E$ then $(j,i) \in E$, hence A is symmetric. This proves that Δ is symmetric.
In addition it is clear that $\bar{D}^{-\frac{1}{2}}$ and $\tilde{D}^{-\frac{1}{2}}$ are diagonal, so we can say that they are symmetric. Hence:

$$\bar{\Delta}^T = (\bar{D}^{-\frac{1}{2}}\Delta\bar{D}^{-\frac{1}{2}})^T = (\bar{D}^{-\frac{1}{2}})^T\Delta^T(\bar{D}^{-\frac{1}{2}})^T = \bar{\Delta}$$

$$\tilde{\Delta}^T = (\tilde{D}^{-\frac{1}{2}}\Delta\tilde{D}^{-\frac{1}{2}})^T = (\tilde{D}^{-\frac{1}{2}})^T\Delta^T(\tilde{D}^{-\frac{1}{2}})^T = \tilde{\Delta}$$

Now, let $X = (x_i)_{i\in\{1,\dots,V\}}$ be a 1-dimensional signal on G:

$$\begin{aligned} X^T\Delta X &= \sum_{i=1}^{V} x_i \sum_{j\sim i} x_i - x_j \\ &= \sum_{i,j=1}^{V} A_{i,j}x_i^2 - x_ix_j \\ &= \sum_{i,j=1}^{V} A_{i,j}\frac{1}{2}(x_i - x_j)^2 + \frac{1}{2}(x_i^2 - x_j^2) \\ &= \frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}(x_i - x_j)^2 \geq 0 \end{aligned}$$

Based on this result, a same result can be found for $\bar{\Delta}$ and $\tilde{\Delta}$:

$$\begin{aligned} X^T\bar{\Delta}X &= X^T\bar{D}^{-\frac{1}{2}}\Delta\bar{D}^{-\frac{1}{2}}X = (\bar{D}^{-\frac{1}{2}}X)^T\Delta(\bar{D}^{-\frac{1}{2}}X) \\ &= \frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}})^2 \geq 0 \end{aligned}$$

$$\begin{aligned} X^T\tilde{\Delta}X &= X^T\tilde{D}^{-\frac{1}{2}}\Delta\tilde{D}^{-\frac{1}{2}}X = (\tilde{D}^{-\frac{1}{2}}X)^T\Delta(\tilde{D}^{-\frac{1}{2}}X) \\ &= \frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}(\frac{x_i}{\sqrt{d_i+1}} - \frac{x_j}{\sqrt{d_j+1}})^2 \geq 0 \end{aligned}$$

Hence, Δ, $\bar{\Delta}$ and $\tilde{\Delta}$ are semi-definite, however, we can prove that they are not definite by considering $X = (1)_{i\in\{1,...,V\}}$ for Δ, $X = (\sqrt{d_i})_{i\in\{1,...,V\}}$ for $\bar{\Delta}$ and $X = (\sqrt{d_i+1})_{i\in\{1,...,V\}}$ for $\tilde{\Delta}$. o.ε.δ.

Theorem 3. — *The eigenvalues of* $\bar{\Delta}$ *are in* $[0, 2]$ *and the eigenvalues of* $\tilde{\Delta}$ *are in* $[0, 2)$

PROOF : Because $\bar{\Delta}$ and $\tilde{\Delta}$ are positive semi-definite (Theorem 1), every eigenvalues are greater than 0. Now, let X be a 1-dimensional signal on G such that $||X||_2 = 1$, then:

$$\begin{aligned} X^T\bar{\Delta}X &= \frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}})^2 \\ &\leq \sum_{i,j=1}^{V} A_{i,j}(\frac{x_i^2}{d_i} + \frac{x_j^2}{d_j}) \\ &= \sum_{i=1}^{V} x_i^2 + \sum_{j=1}^{V} x_j^2 \leq 2 \end{aligned}$$

$$\begin{aligned} X^T\tilde{\Delta}X &= \frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}(\frac{x_i}{\sqrt{d_i+1}} - \frac{x_j}{\sqrt{d_j+1}})^2 \\ &\leq \sum_{i,j=1}^{V} A_{i,j}(\frac{x_i^2}{d_i+1} + \frac{x_j^2}{d_j+1}) \\ &= \sum_{i=1}^{V} \frac{d_i}{d_i+1}x_i^2 + \sum_{j=1}^{V} \frac{d_j}{d_j+1}x_j^2 < 2 \end{aligned}$$

Let λ be the largest eigenvalue of $\bar{\Delta}$, and let X be the associated eigenvector, then $X^T\bar{\Delta}X = \lambda X^T X = \lambda \leq 2$.

Let λ be the largest eigenvalue of $\tilde{\Delta}$, and let X be the associated eigenvector, then $X^T\tilde{\Delta}X = \lambda X^T X = \lambda < 2$. o.$\varepsilon$.$\delta$.

Definition 6 (Pseudo inner-product). Let a S be a symmetric positive semi-definite matrix, the mapping $<,>_S$: $(X, Y) \in \mathcal{X}^l(G)^2 \longmapsto tr(X^T SY) \in \mathbb{R}$ is the pseudo inner-product associated with the matrix S.

Theorem 4. — *Pseudo inner-product associated with the matrix S is indeed a pseudo inner-product.*

PROOF : Let X, Y, Z be l-dimensional signal on G, then:

- **Symmetry**:

$$< X, Y >_S = tr(X^T SY) = tr((X^T SY)^T) = tr(Y^T SX) =< Y, X >_S$$

- **Linearity**: Let $a, b \in \mathbb{R}$, then:

$$\begin{aligned} < aX + bY, Z >_S &= tr((aX + bY)^T SZ) = a \times Tr(X^T SZ) + b \times Tr(Y^T SZ) \\ &= a < X, Z >_S + b < Y, Z >_S \end{aligned}$$

- **Positive semi-definiteness**: By applying the fact that S is a positive semi-definite matrix it follows that:

$$< X, X >_S = tr(X^T SX) = \sum_{k=1}^{V} tr(X_{.,k}^T SX_{.,k}) \geq 0 \qquad \text{o.}\varepsilon\text{.}\delta\text{.}$$

By combining the definition of the pseudo inner product associated with a symmetric, positive, semi-definite matrix and the results from theorem 2, we can consider in the rest of the report the following pseudo inner products: $< ., . >_\Delta$, $< ., . >_{\bar{\Delta}}$ and $< ., . >_{\tilde{\Delta}}$.

Hence we have defined on $\mathcal{X}^l(G)$ three different pseudo-Euclidean structures. We will prove that those spaces are very different but that they are useful for studying different measures of over-smoothing on a graph.

4.3 Fourier Transform on Graphs

It will be useful in the rest of the paper to be able to perform the Fourier transform of a signal $X \in \mathcal{X}^l(G)$ associated with $S \in \{\Delta, \bar{\Delta}, \tilde{\Delta}\}$. By theorem 3, we know that the eigenvalues of S are positive, hence we can order them:

$$0 = \lambda_1 \leq ... \leq \lambda_V$$

In (Chung [4]), it is proved that $\lambda_2 = 0$ if and only if G is not connected, hence $\lambda_2 > 0$ in our case.

There exist a square matrix P such that $S = P^{-1} diag(\lambda_1, ..., \lambda_V) P$.

Definition 7 (Fourier transform on graphs). If X is a l-dimensional signal on G, the Fourier transform of X is:

$$\widehat{X} = PX$$

The Fourier transform is build so that the following diagram is commutative:

$$\begin{array}{ccc} \mathcal{X}^l(G) & \xrightarrow{\quad S \quad} & \mathcal{X}^l(G) \\ \Big\downarrow{\scriptstyle P} & & \Big\uparrow{\scriptstyle P^{-1}} \\ \widehat{\mathcal{X}}^l(G) & \xrightarrow[diag(\lambda_1,\dots,\lambda_V)]{} & \widehat{\mathcal{X}}^l(G) \end{array}$$

Definition 8. Let $M : \mathcal{X}^l(G) \mapsto \mathcal{X}^l(G)$, then the Fourier transform of M is $\widehat{M} = P \circ M \circ P^{-1}$. In particular, if M is a squared matrix of dimension V, $\widehat{M} = PMP^{-1}$

This definition assures us that applying M in $\mathcal{X}^l(G)$ is equivalent to applying $\widehat{M}$ in $\widehat{\mathcal{X}}^l(G)$. Let $X \in \mathcal{X}^l(G)$, then:

$$\widehat{M}\widehat{X} = P \circ M \circ P^{-1}PX = PMX = \widehat{MX}$$

4.4 Dirichlet Energy

Definition 9 (Dirichlet Energy). Let $X \in \mathcal{M}_{V\times l}(\mathbb{R})$ be a l-dimensional signal on the graph G. The Dirichlet energy of X associated with Δ, $\bar{\Delta}$ or $\tilde{\Delta}$ are:

$$E_{\Delta}(X) = \sqrt{\frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}\,||X_i - X_j||_2^2}$$

$$E_{\bar{\Delta}}(X) = \sqrt{\frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}\left|\left|\frac{X_i}{\sqrt{d_i}} - \frac{X_j}{\sqrt{d_j}}\right|\right|_2^2}$$

$$E_{\tilde{\Delta}}(X) = \sqrt{\frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}\left|\left|\frac{X_i}{\sqrt{d_i+1}} - \frac{X_j}{\sqrt{d_j+1}}\right|\right|_2^2}$$

Theorem 5. — *The Dirichlet energy associated with* $S \in \{\Delta, \bar{\Delta}, \tilde{\Delta}\}$ *is the pseudo norm associated to* $< ., . >_S$

PROOF :

$$\begin{aligned} E_\Delta(X)^2 &= \frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}\,||X_i - X_j||_2^2 \\ &= \frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}\sum_{k=1}^{V} x_{i,k} - x_{j,k})^2 \\ &= \sum_{k=1}^{V}(\frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}(x_{i,k}x_{j,k})^2) \\ &= \sum_{k=1}^{V} tr(X_{.,k}^T \Delta X_{.,k}) \\ &= tr(X^T \Delta X) =< X, X >_\Delta \end{aligned}$$

$$\begin{aligned} E_{\bar{\Delta}}(X)^2 &= \frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}\left|\left|\frac{X_i}{\sqrt{d_i}} - \frac{X_j}{\sqrt{d_j}}\right|\right|_2^2 \\ &= \frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}\sum_{k=1}^{V}(\frac{x_{i,k}}{\sqrt{d_i}} - \frac{x_{j,k}}{\sqrt{d_j}})^2 \\ &= \sum_{k=1}^{V}(\frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}(\frac{x_{i,k}}{\sqrt{d_i}} - \frac{x_{j,k}}{\sqrt{d_j}})^2) \\ &= \sum_{k=1}^{V} tr(X_{.,k}^T \bar{\Delta} X_{.,k}) \\ &= tr(X^T \bar{\Delta} X) =< X, X >_{\bar{\Delta}} \end{aligned}$$

$$
\begin{aligned}
E_{\tilde{\Delta}}(X)^2 &= \frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}\left|\left|\frac{X_i}{\sqrt{d_i+1}} - \frac{X_j}{\sqrt{d_j+1}}\right|\right|_2^2 \\
&= \frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}\sum_{k=1}^{V}(\frac{x_{i,k}}{\sqrt{d_i+1}} - \frac{x_{j,k}}{\sqrt{d_j+1}})^2 \\
&= \sum_{k=1}^{V}(\frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}(\frac{x_{i,k}}{\sqrt{d_i+1}} - \frac{x_{j,k}}{\sqrt{d_j+1}})^2) \\
&= \sum_{k=1}^{V} tr(X_{.,k}^T\tilde{\Delta}X_{.,k}) \\
&= tr(X^T\tilde{\Delta}X) =< X, X >_{\tilde{\Delta}} \qquad \text{o.}\varepsilon\text{.}\delta\text{.}
\end{aligned}
$$

Definition 10. Let X be a l-dimensional signal on G: $\widehat{E_S}(\widehat{X}) = \sqrt{tr(\widehat{X}^T diag(\lambda_1, ..., \lambda_V)\widehat{X})}$

Theorem 6. — *Let X be a l-dimensional signal on G:*

$$\widehat{E_S}(\widehat{X}) = E_S(X)$$

PROOF :

$$
\begin{aligned}
tr(\widehat{X}^T diag(\lambda_1, ..., \lambda_V)\widehat{X}) &= tr(PX^TP^{-1}diag(\lambda_1, ..., \lambda_V)PXP^{-1}) \\
&= tr(X^TSXP^{-1}P) \\
&= tr(X^TSX) \qquad \text{o.}\varepsilon\text{.}\delta\text{.}
\end{aligned}
$$

Moreover, using the Fourier transform, we have a very nice formula for the squared Dirichlet energy. Let X be a 1-dimensional signal on G, then:

$$E_S^2(X) = \sum_{i=1}^{V} \lambda_i \widehat{x_i}^2$$

We call the eigenvalues $\lambda_1, ..., \lambda_V$ the frequencies of the graph G. The energy of a signal X directly depends of the frequencies that compose it.

Theorem 7. — *It is possible to obtain a type of Cauchy Schwartz inequality for the Dirichlet energy. Let $S \in \{\Delta, \bar{\Delta}, \tilde{\Delta}\}$, then for $X, Y \in \mathcal{X}^l(G)$:*

$$|< X, Y >_S| \leq E_S(X)E_S(Y)$$

PROOF : The proof of this inequality is very similar to the proof of the Cauchy-Schwartz inequality. Let $X, Y \in \mathcal{X}^l(G)$ and $P : t \in \mathbb{R} \mapsto < tX + Y, tX + Y >_S$. Then, $P(t) = t^2 E_S^2(X) + 2t < X, Y >_S + E_S^2(Y)$ is a polynomial of degree two that have at most one solution. Hence, $4 < X, Y >_S^2 -4E_S^2(X)E_S^2(Y) \leq 0$, so:

$$|< X, Y >_S| \leq E_S(X)E_S(Y) \qquad \text{o.}\varepsilon\text{.}\delta\text{.}$$

Definition 11 (Matrix Norm associated to the Dirichlet Energy). Let $S \in \{\Delta, \bar{\Delta}, \tilde{\Delta}\}$, we define the matrix pseudo-norm associated to E_S for a matrix M as :

$$||M||_S = \sup_{E_S(X)=1} E_S(MX) \in \mathbb{R}^+ \cup \{+\infty\}$$

The positivity and the triangular inequality of this pseudo-norm result from the positivity and the triangular inequality of E_S.

It is then clear that for a matrix M and signal X that we have:

$$E_S(MX) \leq ||M||_S \, E_S(X)$$

Theorem 8. — *Let P be the graph Fourier transform associated with S. $||M||_S < +\infty$ if and only if the eigenvector of S associated to 0 is an eigenvector of M.*

PROOF : Let X be a 1-dimensional signal on G.

$$\begin{aligned} E_S^2(MX) &= \widehat{E}_S^2(\widehat{MX}) = \widehat{E}_S^2(\widehat{M}\widehat{X}) \\ &= \sum_{i=1}^{V} \lambda_i (\widehat{M}\widehat{X})_i^2 \end{aligned}$$

If $\widehat{X} = (\widehat{X}_1, ..., \widehat{X}_V)^T$, then $\widehat{X}' = (\widehat{X}_2, ..., \widehat{X}_V)^T$.

If $\widehat{M} = (a_{i,j})_{i,j \in \{1,...,V\}}$ then $\widehat{M}' = (a_{i,j})_{i,j \in \{2,...,V\}}$

1. Suppose that the eigenvector of S associated to 0 is an eigenvector of M, and that $E_S(X) = 1$. The first condition assures us that $[1, 0, ..., 0]^T$ is an

eigenvector of $\widehat{M}$, hence:

$$\begin{aligned} E_S^2(MX) &= \sum_{i=1}^{V} \lambda_i (\widehat{M}\widehat{X})_i^2 \\ &= \sum_{i=2}^{V} \lambda_i (\widehat{M}'\widehat{X}')_i^2 \\ &\leq \lambda_{max} \sum_{i=2}^{V} (\widehat{M}'\widehat{X}')_i^2 \\ &= \lambda_{max} \left|\left|\widehat{M}'\widehat{X}'\right|\right|_2^2 \\ &= \lambda_{max} \left|\left|\widehat{M}'\right|\right|_2^2 \left|\left|\widehat{X}'\right|\right|_2^2 \end{aligned}$$

$E_S^2(X) = 1$ implies that for $i \in \{2, ..., V\}$ $\widehat{X}_i^2 \leq \dfrac{1}{\lambda_i}$, hence

$$\left|\left|\widehat{X}'\right|\right|_2^2 \leq \sum_{i=2}^{V} \frac{1}{\lambda_i} := C^2$$

This proves that:

$$||M||_S \leq \sqrt{\lambda_{max}} \left|\left|\widehat{M}'\right|\right|_2 C$$

2. Define the series $(\widehat{X}_n)_{n\in\mathbb{N}}$ as $\widehat{X}_n = \left(n, \frac{1}{\sqrt{\lambda_2(V-1)}}, ..., \frac{1}{\sqrt{\lambda_V(V-1)}}\right)^T$. We see that:

$$E_S^2(X_n) = \widehat{E}_S^2(\widehat{X}_n) = 0 \times n + \sum_{i=2}^{V} \lambda_i \left(\frac{1}{\sqrt{\lambda_i(V-1)}}\right)^2 = 1$$

Suppose the eigenvector of S associated to 0 is not an eigenvector of M, hence $v_1 = [1, 0, ..., 0]^T$ is not an eigenvector of $\widehat{M}$. Let $v_1 = [1, 0, ..., 0]^T, v_2 = [0, 1, ..., 0]^T ..., v_V = [0, 0, ..., 1]^T$, then there exist $k \in \{2, ..., V\}$ such that $\left\langle \widehat{M}v_1, v_k \right\rangle := \alpha \neq 0$.

$$E_S^2(MX_n) \geq \lambda_k (\widehat{M}\widehat{X}_n)_k^2$$

Because the only coordinate that depends on n is the first one, there exist β such that:

$$\lambda_k(\widehat{M}\widehat{X}_n)_k^2 = \lambda_k(\alpha n + \beta)^2 \xrightarrow{n\longrightarrow+\infty} +\infty$$

Hence $||M||_S = +\infty$ o.ε.δ.

Theorem 9. — *We have the following relations:*

$$\begin{aligned} E_{\bar{\Delta}} &= E_{\Delta} \circ D^{\frac{1}{2}} \\ E_{\tilde{\Delta}} &= E_{\Delta} \circ \tilde{D}^{\frac{1}{2}} \\ E_{\tilde{\Delta}} &= E_{\bar{\Delta}} \circ D^{-\frac{1}{2}} \circ \tilde{D}^{\frac{1}{2}} \end{aligned}$$

PROOF : Let $X \in \mathcal{X}^l(G)$:

$$\begin{aligned} E^2_{\bar{\Delta}}(X) &= tr(X^T D^{-\frac{1}{2}} \Delta D^{\frac{1}{2}} X) \\ &= tr((D^{-\frac{1}{2}} X)^T \Delta (D^{\frac{1}{2}} X)) \\ &= E^2_{\Delta}(D^{\frac{1}{2}} X) \end{aligned}$$

$$\begin{aligned} E^2_{\tilde{\Delta}}(X) &= tr(X^T \tilde{D}^{-\frac{1}{2}} \Delta \tilde{D}^{\frac{1}{2}} X) \\ &= tr((\tilde{D}^{-\frac{1}{2}} X)^T \Delta (\tilde{D}^{\frac{1}{2}} X)) \\ &= E^2_{\Delta}(\tilde{D}^{\frac{1}{2}} X) \end{aligned}$$

$$\begin{aligned} E_{\tilde{\Delta}} &= E_{\Delta} \circ \tilde{D}^{\frac{1}{2}} \\ &= E_{\bar{\Delta}} \circ D^{-\frac{1}{2}} \circ \tilde{D}^{\frac{1}{2}} \end{aligned}$$ o.ε.δ.

These formulas enable us to show that the pseudo Euclidean spaces associated with each Laplacian are very different.

Theorem 10. — *The norms $E_{\Delta}, E_{\bar{\Delta}}$ and $E_{\tilde{\Delta}}$ are not equivalent on a non-regular graph.*

PROOF : Let $v_1 = (1)_{i\in\{1,\dots,V\}}$, $v_2 = (\sqrt{d_i})_{i\in\{1,\dots,V\}}$ and $v_3 = (\sqrt{d_i+1})_{i\in\{1,\dots,V\}}$, because the graph is not regular, v_1, v_2 and v_3 are pair-wise not co-linear. These vectors are the eigenvectors associated with 0 of Δ, $\bar{\Delta}$ and $\tilde{\Delta}$.
Moreover, $D^{\frac{1}{2}} v_1 = v_2$ and $\tilde{D}^{\frac{1}{2}} v_1 = v_3$, hence by the theorem 8, we can deduce that E_{Δ} and $E_{\bar{\Delta}}$ are not co-linear and that E_{Δ} and $E_{\tilde{\Delta}}$. Similarly, it is easy to show that v_2 is not an eigenvector of $D^{-\frac{1}{2}} \circ \tilde{D}^{\frac{1}{2}}$, which concludes the proof. o.ε.δ.

5 A mathematical approach of over-smoothing

Several papers give different definitions of over-smoothing. We will use the definition introduced in (Rusch et al. [8]) and prove that the Dirichlet Energy introduced earlier is indeed a vertex similarity measure. we will also extend the notion of over-smoothing to Diffusion processes on graphs and show that the exponential convergence of the Dirichlet energy is the right bound.

Moreover, we will show that only in the case of Graph Convolution Networks we can have a bound for the Dirichlet energy. Indeed, as soon as we consider Residual Graph Neural Networks, it is not possible.

5.1 A tractable definition of over-smoothing

Definition 12 (Vertex similarity measure). Let X be a l-dimensional signal on the graph G, a vertex similarity measure is a function $\mu : X \in \mathcal{X}^l(G) \mapsto \mathbb{R}_+$ such that:

1. $\mu(X) = 0$ if and only if there exist a feature vector $c \in \mathbb{R}^l$ such that $X = (c)_{i\in\{1,...,V\}}$ (*i.e.* every vertex share the same features).
2. $\mu(X + Y) \leq \mu(X) + \mu(Y)$

Theorem 11. — *The mapping $\mu : X \mapsto E_\Delta(X)$ is a vertex similarity measure.*

Proof : 1. Let X be an l-dimensional signal on the graph G such that $\mu(X) = 0$, this means that $E^2_\Delta(X) = 0$, hence:

$$tr(X^T \Delta X) = 0$$
$$\frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}||X_i - X_j||_2^2 = 0$$

This proves that for all $(i, j) \in E$, $X_i = X_j$, because G is connected this concludes that there exists $c \in \mathbb{R}^l$ such that $X_i = c$ for all $i \in \{1, ..., V\}$. Moreover if this condition is respected it is clear that $E^2_\Delta(X) = \frac{1}{2}\sum_{i,j=1}^{V} A_{i,j}||X_i - X_j||_2^2 = 0$.

2.

$$\begin{aligned}\mu(X+Y)^2 =& < X+Y, X+Y >_\Delta \\ =& < X, X >_\Delta + 2 < X, Y >_\Delta + < Y, Y >_\Delta \\ \leq& < X, X >_\Delta + 2| < X, Y >_\Delta | + < Y, Y >_\Delta \\ \leq& < X, X >_\Delta + 2\sqrt{< X, X >_\Delta}\sqrt{< Y, Y >_\Delta} + < Y, Y >_\Delta \\ =& \mu(X)^2 + 2\mu(X)\mu(Y) + \mu(Y)^2 \\ =& (\mu(X) + \mu(Y))^2\end{aligned}$$

Hence, $\mu(X+Y) \leq \mu(X) + \mu(Y)$ o.ε.δ.

We can slightly expand the notion of vertex similarity measure, only E_Δ exactly verifies this definition because of the first condition in the definition. If we keep only the second condition, then $E_{\tilde{\Delta}}$ and $E_{\tilde{\Delta}}$ can be considered vertex similarity measures. The difference is that the signals that minimize the energy are no longer equal on all vertices but they still are the eigenvectors of $\lambda_1 = 0$.

Definition 13 (Over-smoothing). We say that a series of l-dimensional signals $(X^{(n)})_{n\geq 0}$ is over-smoothing with respect to a vertex similarity measure μ if $\mu(X^{(n)}) = O(\lambda^n)$ for some $0 < \lambda < 1$.

Similarly, we say that $(X(t))_{t\geq 0}$ is over-smoothing if there exist $0 < \lambda < 1$ such that $\mu(X(t)) = O(\lambda^t)$.

5.2 Graph Convolution Networks

The Graph Convolution Networks was introduced in (Kipf et al. [7]) and follows the following update rule:

$$X^{(k+1)} = \sigma((...(\sigma((\mathbb{I}d - \tilde{\Delta})X^{(k)}W_{k,1})W_{k,2})...)W_{k,m})$$

N.B: $\sigma(x) = max(0, x)$

This definition is indeed in accordance with the classification of the different Graph Neural Network architectures. For $i \in \{1, ..., V\}$:

$$(\tilde{\Delta}(X^{(k)}))_i = X_i^{(k)} - \sum_{j\sim i} X_j^{(k)} \frac{\sqrt{d_i + 1}}{\sqrt{d_j + 1}}$$

Hence by defining:

$$\begin{aligned} H_{i,j} &= \frac{\sqrt{d_i + 1}}{\sqrt{d_j + 1}} \\ \gamma^{(k)}(a, b) &= \sigma((...(\sigma(b)W_{k,1})W_{k,2})...)W_{k,m}) \\ \bigoplus_{j \sim i} X_j &= \sum_{j \sim i} X_j \end{aligned}$$

We find that:

$$(\tilde{\Delta}(X^{(k)}))_i = \gamma^{(k)} \left(X_i^{(k)}, \bigoplus_{j \sim i} H_{i,j} X_j^{(k)} \right)$$

In *A note on over-smoothing for graph neural networks* (Cai et al. [1]), the following theorem is proved:

Theorem 12. — *Suppose that the for all k and $l \leq m$, $\lambda_{max}(W_{k,m}W_{k,m}^T) < 1$, then $E_{\tilde{\Delta}}(X^{(k)}) \leq (1 - \lambda_2(\tilde{\Delta}))^k E_{\tilde{\Delta}}(X^{(k)})$.*

This result is a very strong result, however, it is not generalizable to Residual Graph Convolution Networks and to Graph Neural Ordinary Differential Equation Networks. Hence to study over-smoothing for those architectures it is necessary to simplify the problem.

In the proof of theorem 12 in (Cai et al. [1]), except from some technical points on the function σ and some condition on the weights matrices $W_{k,l}$, the important point revolves around the dynamics of $X \mapsto (\mathbb{I}d - \tilde{\Delta})X$ and how the Dirichlet Energy evolves under this dynamic. In the rest, we will study the continuous version of this dynamic.

5.3 Over-smoothing and isotropic diffusion

Let $S \in \{\Delta, \bar{\Delta}, \tilde{\Delta}\}$. We can view the dynamic $X \mapsto (\mathbb{I}d - S)X$ as an Euler discretization scheme of the following process:

$$\dot{X}(t) = -SX(t) \tag{3}$$

where $X(t)$ is a 1-dimensional signal on the graph G. We can prove that the Dirichlet energy converge to 0 exponentially fast. By using the theorem 6, we have:

$$\begin{aligned}
\frac{dE_S^2(X(t))}{dt} &= \frac{d\widehat{E}_S^2(\widehat{X}(t))}{dt} \\
&= 2 < \widehat{X}(t), \frac{d\widehat{X}(t)}{dt} >_S \\
&= -2 < \widehat{X}(t), \widehat{S}\widehat{X}(t) >_{\widehat{S}} \\
&= -2 \sum_{i=1}^{V} \lambda_i^2 X_i(t)^2 \\
&\leq -2\lambda_2 \sum_{i=1}^{V} \lambda_i X_i(t)^2 \\
&= -2\lambda_2 E_S^2(X(t))
\end{aligned}$$

Similarly, we obtain:

$$\frac{dE_S^2(X(t))}{dt} \geq -2\lambda_V E_S^2(X(t))$$

Hence we have shown that the Dirichlet Energy converge exactly at an exponential rate to 0 and that this rate depends on the frequencies of the graph G.

The diffusion defined in equation 3 is called the isotropic diffusion for a reason that we will see in the next part.

6 Diffusion on graphs

The links between equation 3 on graphs and the formalism to a general form of anisotropic diffusion on graphs is based on *GRAND: Graph Neural Diffusion* (Chamberlain et al. [2]). In this part, we generalize the notion of diffusion on graphs to the three Laplacians, $\Delta, \bar{\Delta}$ and $\tilde{\Delta}$ and we study how the Dirichlet energy evolves with time.

6.1 The diffusion equation

Let $\mathcal{X}^l(G) = \mathcal{M}_{V\times l}(\mathbb{R})$ be the set of all l-dimensional vertex signals on the graph G. Let $\mathcal{H}^l(G)$ be the set of all l-dimensional edge signals on G. We define an edge signal $\epsilon \in \mathcal{H}^l(G)$ as a function $(i,j) \in \{1,...,V\}^2 \mapsto \epsilon(i,j) \in \mathbb{R}^l$ such that:

1. $(i,j) \notin E \implies \epsilon(i,j) = 0$
2. $\epsilon(i,j) = -\epsilon(j,i)$

We can equip $\mathcal{X}^l(G)$ and $\mathcal{H}^l(G)$ with an inner product. Let $X, Y \in \mathcal{X}^l(G)$, and $h, g \in \mathcal{H}^l(G)$

$$\langle X, Y \rangle = tr(X^T Y)$$
$$\ll h, g \gg = \frac{1}{2} \sum_{i,j=1}^{V} A_{i,j} \langle h(i,j), g(i,j) \rangle$$

In addition, we can define a gradient operator $\nabla : \mathcal{X}^l(G) \longrightarrow \mathcal{H}^l(G)$ and a divergence operator $div : \mathcal{H}^l(G) \longrightarrow \mathcal{X}^l(G)$:

- For $(i,j) \in \{1,...,V\}^2$ and $X \in \mathcal{X}^l(G)$, $(\nabla X)_{i,j} = X_i - X_j$ if $(i,j) \in E$, otherwise $(\nabla X)_{i,j} = 0$
- For $i \in \{1,...,V\}$ and $\epsilon \in \mathcal{H}^l(G)$, $div(\epsilon)_i = \sum_{j\sim i} \epsilon(i,j)$

Theorem 13. — *The ∇ operator and the div operator are adjoint:*

$$\langle x, div(h) \rangle = \ll \nabla x, h \gg$$

Let $M : \mathcal{X}^l(G) \times \mathbb{R}_+ \longrightarrow \mathcal{M}_{V\times V}(\mathbb{R})$, such that $M(x(t),t)_{i,i} = 1$ for every $i \in \{1,...,V\}$ and $M(x(t),t)_{i,j} = 0$ if $(i,j) \notin E$. It makes sense to consider the following equation:

$$\dot{X}(t) = div(M(X(t),t)\nabla X) \tag{4}$$

We call the equation 4 the diffusion equation on the graph G.

Theorem 14. — *By seeing the Laplacian matrix Δ as a mapping from $\mathcal{X}^l(G)$ to itself, then $\Delta = div(\nabla)$*

PROOF : Let X be an l-dimensional signal on G.

$$\begin{aligned} div(\nabla X) &= div((A_{i,j}(X_i - X_j))_{i,j\in\{1,...,V\}}) \\ &= (\sum_{j=1}^{V} A_{i,j}A_{i,j}(X_i - X_j))_{i\in\{1,...,V\}} \\ &= (\sum_{j=1}^{V} A_{i,j}(X_i - X_j))_{i\in\{1,...,V\}} \\ &= \Delta(X) \end{aligned}$$ o.ε.δ.

The definition the gradient operator and the divergence operator can also be modified to obtain the normalized augmented Laplacian $\tilde{\Delta}$.

Consider $\tilde{\nabla} : \mathcal{X}^l(G) \longrightarrow \mathcal{H}^l(G)$ and $\tilde{div} : \mathcal{H}^l(G) \longrightarrow \mathcal{X}^l(G)$:

- For $(i,j) \in \{1,...,V\}^2$ and $X \in \mathcal{X}^l(G)$, $(\tilde{\nabla}X)_{i,j} = \frac{X_i}{\sqrt{d_i+1}} - \frac{X_j}{\sqrt{d_j+1}}$ if $(i,j) \in E$, otherwise $(\tilde{\nabla}X)_{i,j} = 0$
- For $i \in \{1,...,V\}$ and $\epsilon \in \mathcal{H}^l(G)$, $\tilde{div}(\epsilon)_i = \frac{1}{\sqrt{d_i+1}} \sum_{j\sim i} \epsilon(i,j)$

Theorem 15. — *By seeing the normalized augmented Laplacian matrix $\tilde{\Delta}$ as a mapping from $\mathcal{X}^l(G)$ to itself, then $\tilde{\Delta} = \tilde{div}(\tilde{\nabla})$*

PROOF : Let X be an l-dimensional signal on G.

$$\begin{aligned} \tilde{div}(\tilde{\nabla}X) &= \tilde{div}((A_{i,j}(\frac{X_i}{\sqrt{d_i+1}} - \frac{X_j}{\sqrt{d_j+1}}))_{i,j\in\{1,...,V\}}) \\ &= (\frac{1}{\sqrt{d_i+1}} \sum_{j=1}^{V} A_{i,j}A_{i,j}(\frac{X_i}{\sqrt{d_i+1}} - \frac{X_j}{\sqrt{d_j+1}}))_{i\in\{1,...,V\}} \\ &= (\frac{1}{\sqrt{d_i+1}} \sum_{j=1}^{V} A_{i,j}(\frac{X_i}{\sqrt{d_i+1}} - \frac{X_j}{\sqrt{d_j+1}}))_{i\in\{1,...,V\}} \\ &= \tilde{\Delta}(X) \end{aligned}$$ o.ε.δ.

Similarly, we can also obtain the normalized Laplacian $\bar{\Delta}$. Consider $\bar{\nabla} : \mathcal{X}^l(G) \longrightarrow \mathcal{H}^l(G)$ and $\bar{div} : \mathcal{H}^l(G) \longrightarrow \mathcal{X}^l(G)$:

- For $(i,j) \in \{1,...,V\}^2$ and $X \in \mathcal{X}^l(G)$, $(\bar{\nabla}X)_{i,j} = \frac{X_i}{\sqrt{d_i}} - \frac{X_j}{\sqrt{d_j}}$ if $(i,j) \in E$, otherwise $(\tilde{\nabla}X)_{i,j} = 0$
- For $i \in \{1,...,V\}$ and $\epsilon \in \mathcal{H}^l(G)$, $\tilde{div}(\epsilon)_i = \frac{1}{\sqrt{d_i}} \sum_{j \sim i} \epsilon(i,j)$

6.2 Anisotropic and isotropic diffusion on the graph

Definition 14 (Isotropic diffusion). We say that the diffusion equation is isotropic when $M = a\mathbb{I}d$ with $a > 0$.

Definition 15 (Anisotropic diffusion). We say that the diffusion equation is anisotropic in every other case.

From the results seen before, we know that the isotropic diffusion leads to the exponential convergence of the Dirichlet energy to 0. We will study how changing the matrix M modify evolution of the Dirichlet energy.

Theorem 16. — *Let M be a symmetric matrix. $div(M\Delta X) = \Delta_M$ with:*

$$\Delta_M = D_M - M$$
$$D_M = diag(\sum_{i=1}^{V} M_{i,1}, ..., \sum_{i=1}^{V} M_{i,V})$$

Proof : Let X be a 1-dimensional signal on G and $i \in \{1,...,V\}$:

$$\begin{aligned}\Delta_M(X) &= D_M X - MX \\ &= (\sum_{j=1}^{V} M_{i,j}X_i - \sum_{i \sim j} M_{i,j}X_j)_{i \in \{1,...,V\}} \\ &= (\sum_{j=1}^{M} M_{i,j}(X_i - X_j))_{i \in \{1,...,V\}}\end{aligned}$$

In addition, we have:

$$\begin{aligned} div(M\nabla X) &= div(M(A_{i,j}(X_i - X_j))_{i,j\in\{1,\dots,V\}}) \\ &= div((M_{i,j}(X_i - X_j))_{i,j\in\{1,\dots,V\}}) \\ &= (\sum_{j=1}^{V} A_{i,j}M_{i,j}(X_i - X_j))_{i\in\{1,\dots,V\}} \\ &= (\sum_{j=1}^{V} M_{i,j}(X_i - X_j))_{i\in\{1,\dots,V\}} \end{aligned}$$

o.ε.δ.

It is important to see that for a 1-dimensional signal X:

$$\begin{aligned} X^T\Delta_M X &= \sum_{i=1}^{V} X_i \sum_{j\sim i} M_{i,j}(X_i - X_j) \\ &= \sum_{i,j=1}^{V} M_{i,j}(X_i^2 - X_iX_j) \\ &= \sum_{i,j=1}^{V} M_{i,j}(\frac{1}{2}(X_i - X_j)^2 - \frac{1}{2}(X_i^2 - X_j^2) \\ &= \frac{1}{2}\sum_{i,j=1}^{V} M_{i,j}(X_i - X_j)^2 - \frac{1}{2}\sum_{i,j=1}^{V} M_{i,j}(X_i^2 - X_j^2) \end{aligned} \tag{5}$$

Let M be a symmetric matrix with positive coefficients, then equation 5 assures us that Δ_M is positive and semi-definite, hence only have positive eigenvalues.

Theorem 17.— *Let $\varepsilon > 0$ and $M : \mathcal{X}^1(G) \times \mathbb{R}_+ \longrightarrow \mathcal{M}_{V\times V}(\mathbb{R}_+)$ be such that if $(i,j) \notin E$ then $M_{i,j} = 0$ and if $(i,j) \in E$ then $M_{i,j} > \varepsilon$. Then the Dirichlet energy of a signal $X : t \mapsto X(t)$ that solves the equation:*

$$\dot{X}(t) = -div(M(X(t),t)\nabla X(t)) \tag{6}$$

converge exponentially fast to 0.
If the matrix M verifies this condition we say that the diffusion equation is positive anisotropic.

PROOF : Following the same method as for the calculation of the Dirichlet energy of the isotropic diffusion $\dot{X}(t) = -\Delta X(t)$, we find that:

$$\frac{dE^2_{\Delta_{M(X(t),t)}}(X(t))}{dt} \leq -2\lambda_2(\Delta_{M(X(t),t)})E^2_{\Delta_{M(X(t),t)}}(X(t))$$

However:

$$\begin{aligned} X(t)^T \Delta_{M(X(t),t)} X(t) &= \sum_{i,j=1}^{V} M_{i,j}(X(t),t)\, ||X_i(t) - X_j(t)||_2^2 \\ &\geq \varepsilon \sum_{i,j=1}^{V} A_{i,j}\, ||X_i(t) - X_j(t)||_2^2 \end{aligned}$$

With this, we can deduce that $\lambda_2(\Delta_{M(X(t),t)}) \geq \varepsilon\lambda_2(\Delta)$ and that $E_{\Delta_{M(X(t),t)}} \geq \sqrt{\varepsilon}E_\Delta$.

Hence:

$$\frac{dE^2_{\Delta_{M(X(t),t)}}(X(t))}{dt} \leq -2\varepsilon\lambda_2(\Delta)E^2_{\Delta_{M(X(t),t)}}(X(t))$$

Which implies:

$$E_{\Delta_{M(X(t),t)}}(X(t)) \leq E_{\Delta_{M(X(0),0)}}(X(0)) \times e^{-\varepsilon\lambda_2(\Delta)t}$$

And finally,

$$E_\Delta(X(t)) \leq \frac{1}{\sqrt{\varepsilon}} E_{\Delta_{M(X(0),0)}}(X(0)) \times e^{-\varepsilon\lambda_2(\Delta)t} \qquad \text{o.}\varepsilon\text{.}\delta.$$

7 GPU implementation and practical analysis

For this report, we implemented the different notions presented. All the code for generating the figures can be found in the following GitHub repository: https://github.com/adrien-lagesse/ICL-Master-Thesis.

Definition 16 (Erdos-Rényi random graphs). Let $V \geq 1$ and $0 < p < 1$, the Erdos-Rényi graph of parameters V and p is a graph $G = (V, E)$ with V vertices and such that for two vertices $i \neq j$, are linked by an edge with a probability p

We used Erdos-Rényi graphs to test our code and to visualize how the Dirichlet energy evolves under the action of a diffusion process.

First of all, it is very important to note that there are major differences between the graph Fourier transform (hence the energy decomposition) depending on what Laplacian is used.

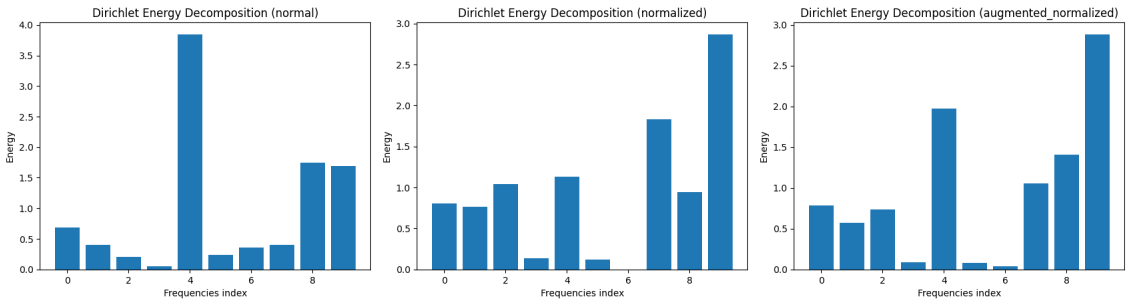

Figure 5: Energy Decomposition by frequencies of a Erdos-Rényi random graph. The Laplacian used is respectively Δ, $\bar{\Delta}$ and $\tilde{\Delta}$

We also proved that the Dirichlet energy converges to 0 for the isotropic diffusion. As a use case, we will consider the dynamics of the normal Laplacian Δ:

$$\dot{X}(t) = -\Delta X(t)$$

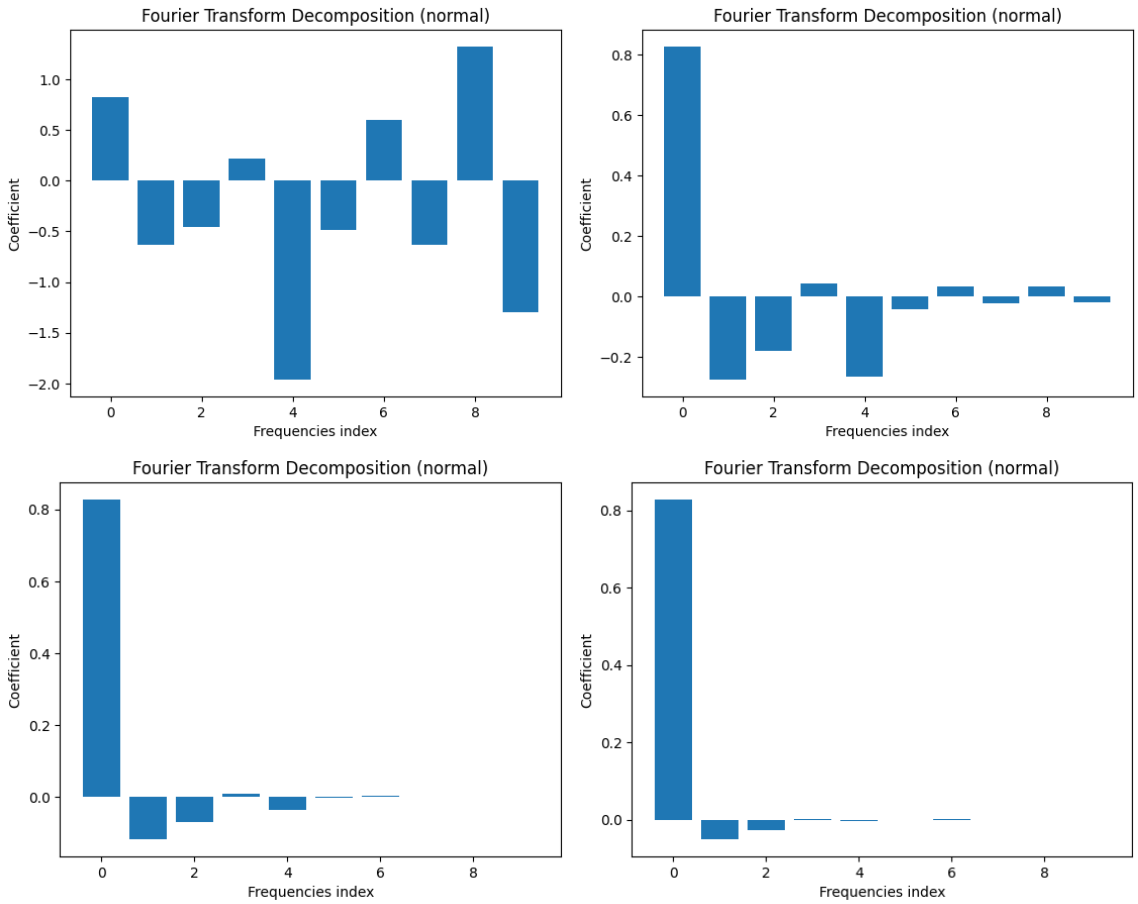

Figure 6: Fourier decomposition of $X(0)$, $X(3.3)$, $X(6.6)$ and $X(10)$

As we can see, the isotropic diffusion acts as a high-frequency filter on the signal X.

In practice, to reduce over-smoothing, we can say that two vertices interact if and only if they are very similar and if two vertices are very different we don't want them to interact together:

$$M(X(t))_{i,j} = \frac{1}{1 + ||X_i(t) - X_j(t)||_2^2}$$

However, when running this diffusion process on a random graph we obtain the results of Figure 7.

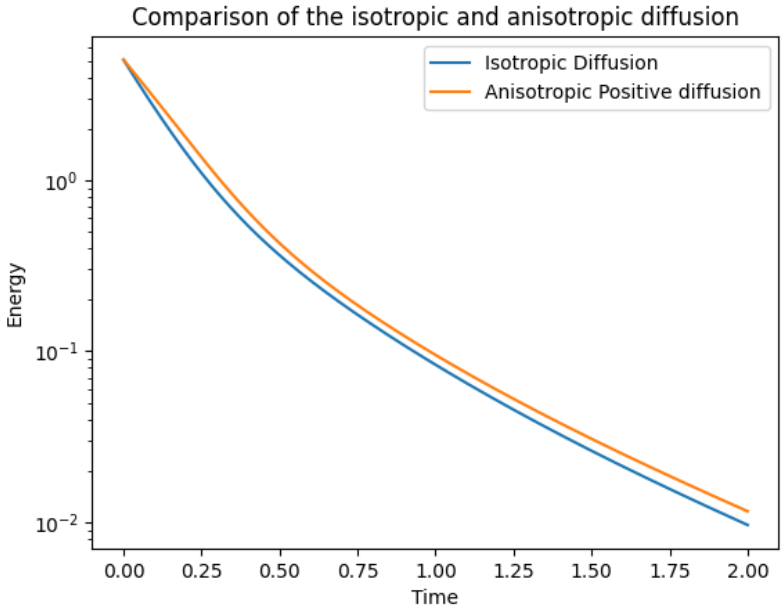

Figure 7: Comparison of the evolution of the Dirichlet Energy for isotropic and anisotropic positive diffusion.

We can apply theorem 17 to indeed prove that even in this case over-smoothing occurs. First let show that under the diffusion $\dot{X}(t) = -div(M(X(t))\nabla X(t))$ the solution $t \mapsto X(t)$ is bounded for the $||.||_2$ norm:

$$\begin{aligned}\frac{d\,||X(t)||_2^2}{dt} &= \frac{\langle X(t), X(t)\rangle}{dt} \\ &= 2\left\langle X(t), \dot{X}(t)\right\rangle \\ &= -2\left\langle X(t), \Delta_{M(X(t))}X(t)\right\rangle \\ &= -2E^2_{\Delta_{M(X(t))}}(X(t)) \leq 0\end{aligned}$$

Hence, because the $||.||_2$ norm is bounded, we can find $\varepsilon > 0$, such that for all $i, j \in \{1, ..., V\}$:

$$\frac{1}{1 + ||X_i(t) - X_j(t)||_2^2} > \epsilon$$

We can now apply theorem 17 and see that the Dirichlet Energy of $X(t)$ converge exponentially fast to 0.

References

[1] Chen Cai and Yusu Wang. *A Note on Over-Smoothing for Graph Neural Networks*. 2020. arXiv: 2006.13318 [cs.LG].

[2] Benjamin Paul Chamberlain et al. *GRAND: Graph Neural Diffusion*. 2021. arXiv: 2106.10934 [cs.LG].

[3] Ricky TQ Chen et al. "Neural ordinary differential equations". In: *Advances in neural information processing systems* 31 (2018).

[4] Fan RK Chung. "Lectures on spectral graph theory". In: *CBMS Lectures, Fresno* 6.92 (1996), pp. 17–21.

[5] Justin Gilmer et al. "Message passing neural networks". In: *Machine learning meets quantum physics* (2020), pp. 199–214.

[6] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[7] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2017. arXiv: 1609.02907 [cs.LG].

[8] T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra. *A Survey on Oversmoothing in Graph Neural Networks*. 2023. arXiv: 2303.10993 [cs.LG].

[9] Franco Scarselli et al. "The Graph Neural Network Model". In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

[10] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL].

[11] Petar Veličković et al. *Graph Attention Networks*. 2018. arXiv: 1710.10903 [stat.ML].