Implementation of "Graph Signal Processing for Directed Graphs based on the Hermitian Laplacian"

The implementation uses GraphWave, but replaces the wavelet used in GraphWave with

$$\boldsymbol{\psi}_{s,i} = \boldsymbol{U}\hat{\boldsymbol{G}}_s\boldsymbol{U}^*\boldsymbol{\delta}_i$$

Here, $\boldsymbol{U}$ is a matrix where the columns are the eigenvectors of the graph Laplacian $\boldsymbol{L}_q$

$$\boldsymbol{L}_q = \boldsymbol{D} - \boldsymbol{\Gamma}_q \odot \boldsymbol{A}^{(s)}$$

Where $\boldsymbol{D}$ is the degree matrix of the symmetrized graph $\boldsymbol{G}^{(s)}$, $\boldsymbol{\Gamma}_q$ is the function

$$\gamma_q(i,j) = \exp(i2\pi q(w_{ij} - w_{ji}))$$

applied to all elements of the adjacency matrix of $\boldsymbol{G}$. $\odot$ is elementwise multiplication. $*$ denotes the conjugate transpose.

$\hat{G}_s = diag(\hat{g}(s\lambda_0), \ldots, \hat{g}(s\lambda_{N-1}))$ where $\hat{g}(s\cdot)$ is a unique filter kernel, in our implementation either a low-pass filter kernel

$$\hat{h}(\lambda) = \frac{1}{1 + c\lambda}$$

or the heat kernel

$$\hat{h}(\lambda) = e^{-s\lambda}$$

Lastly, $\boldsymbol{\delta}_i$ is a vector whose i-th entry is 1 and the others are 0.

After this, we have the wavelets $\boldsymbol{\psi}$. After this, the authors use the same embedding technique that is used by graphwave:

Given a vector T of d values and a vector S of m values, the embeddings are given by

$$x_i = [Re(\phi_i(s,t)), Im(\phi_i(s,t))]_{t\in T, s\in S}$$

Where

$$\phi_i(s,t) = \frac{1}{N}\sum_{j=1}^{N} e^{it\psi_{ij}(s)}$$

This means that phi needs to be calculated for a bunch of s values.

Source: Springer link


```
FURUTANI, Satoshi, et al.
Graph signal processing for directed graphs based on the Hermitian
Laplacian.
In: Joint European Conference on Machine Learning and Knowledge Discovery
in Databases.
Springer, Cham, 2019. p. 447-463.
```