Q& A

Page 3, 7 위주로 살펴보시면 될 듯 합니다.

$$w * h = \mathcal{F}^{-1}(\mathcal{F}(w) \odot \mathcal{F}(h))$$

How to define Fourier transforms for graphs?

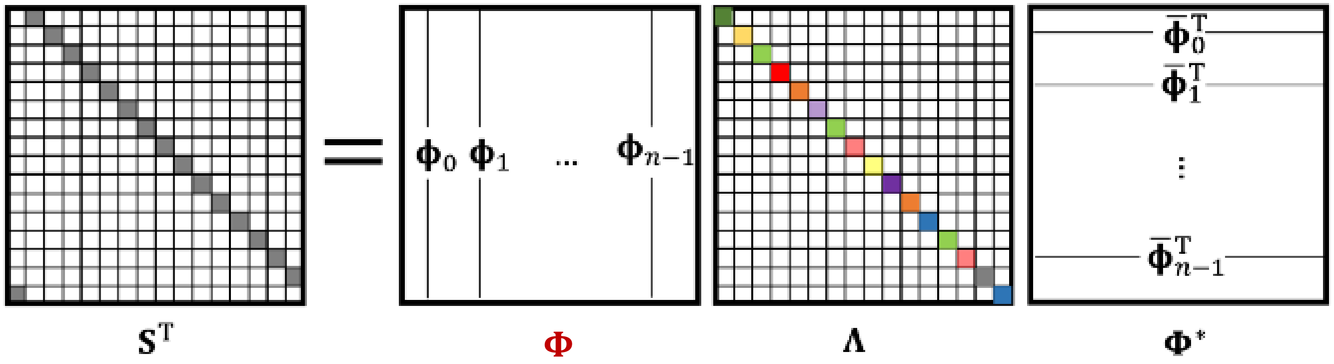
- Rethinking the convolution on sequences
  - Circulant matrices commute! (교환법칙 성립)
  - Commuting matrices are jointly diagonalizable(대각화 가능)

$$AX = \lambda X$$

Eigenvectors (고유벡터) → X; Eigenvalues (고유값) → λ

(eigenvalues는 다를 수 있다)

"모든 circulant matrices의 eigenvectors는 discrete fourier basis와 동일하다"



Shift matirx: $C([0,1\ 0,0,\cdots,0,0])$ ***Eigenvectors*** *Eigenvalues*

✓ 오른쪽으로 1만큼 이동(translation)
✓ Circulant matrix & orthogonal matrix ($S \cdot S^T = I$)

$$\Phi = [e^{(0)}\ e^{(1)}\ e^{(2)}\ \cdots\ e^{(n-1)}]$$

$$e^{(k)} = \begin{bmatrix} w_n^{0\cdot k} \\ w_n^{1\cdot k} \\ w_n^{2\cdot k} \\ \vdots \\ w_n^{(n-1)\cdot k} \end{bmatrix}, \quad w_n = e^{(\frac{2\pi i}{n})}$$

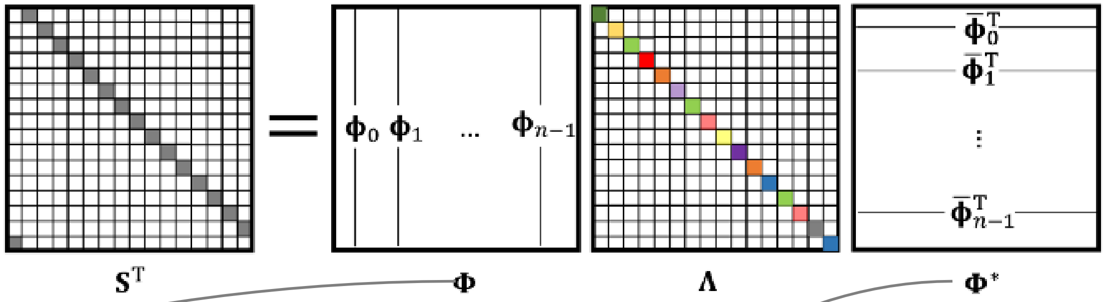
"Discrete Fourier basis"

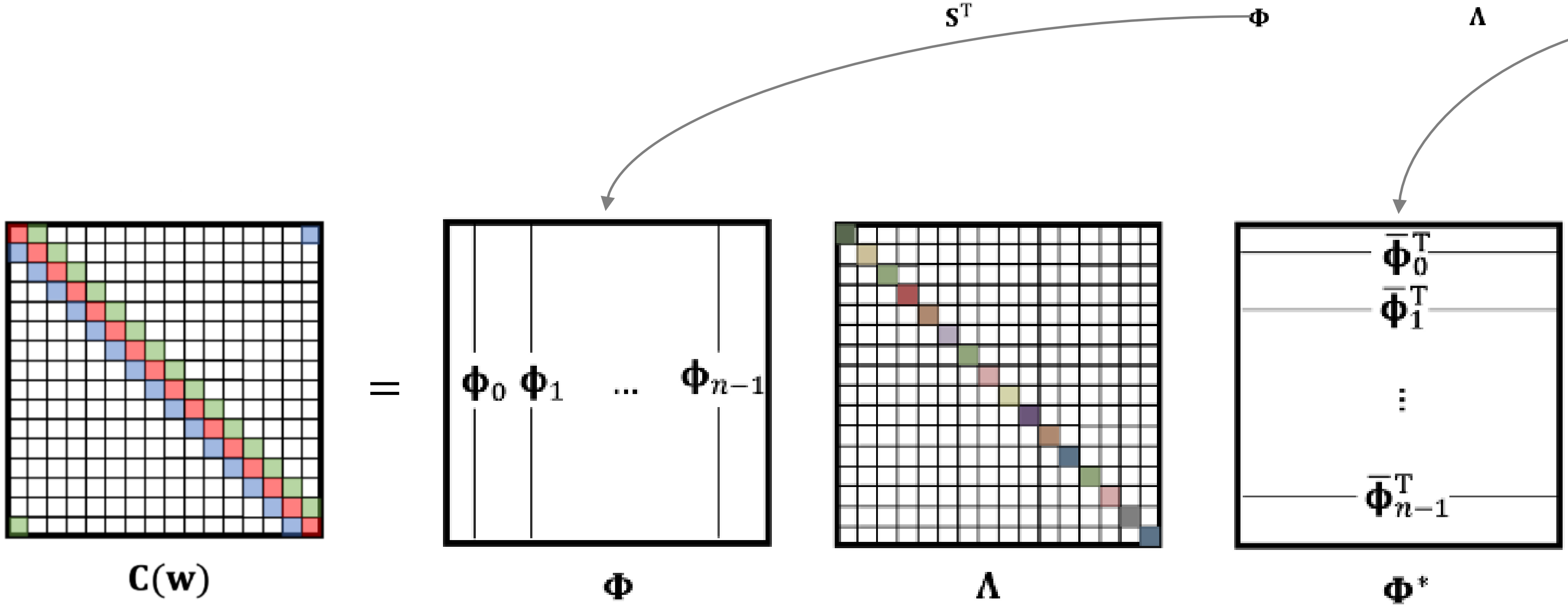
Shift= Translation

Fourier functions form an orthonormal basis

https://towardsdatascience.com/deriving-convolution-from-first-principles-4ff124888028

How to define Fourier transforms for graphs?





- S^T는 translation을 표현하는 가장 단순한(혹은 가장 작은 단위의) circulant matrix
- 모든 circulant matrices는 교환 법칙을 만족하므로 각각의 eigenvectors는 discrete fourier basis와 동일
- $C(w) = C([b, c, 0, 0, \cdots, 0, a])$: a, b, c 를 weight로 하는 circulant matrix(weight가 있는 임의의 회전행렬)
- Φ, Φ^*는 S^T $(shift\ matrix)$에서 가져옴
- 실제로 학습 상황에서는 $C(w)$ 를 모름. 즉, 학습해야할 weight
- 이 때, (고유값 분해 하지 않고) S^T에서 Φ, Φ^* 사용 가능하므로 $C(w)$를 전체 학습하는 것이 아니라 Λ만 학습하면 됨

https://towardsdatascience.com/deriving-convolution-from-first-principles-4ff124888028

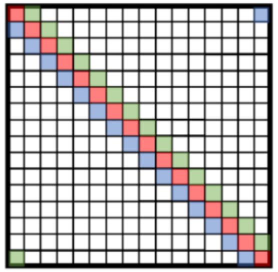
How to define Fourier transforms for graphs?

$$w * h = \mathcal{F}^{-1}(\mathcal{F}(w) \odot \mathcal{F}(h))$$

모르는 값 아는 값 아는 값 모르는 값 아는 값

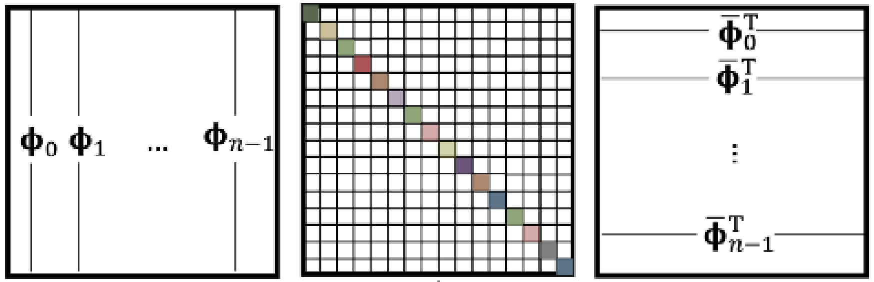
$$\mathcal{F}(h) = \boldsymbol{\Phi}^* \boldsymbol{h}$$

모르는 값을 학습시키자!



$C(w)$

$\mathbf{X}$(Signal)



$\boldsymbol{\Lambda}$(learning parameters)

$$f(X) = \begin{bmatrix} b & c & & & a \\ a & b & c & & \\ & \ddots & \ddots & \ddots & \\ & & a & b & c \\ c & & & a & b \end{bmatrix} \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_{n-1} \\ \mathbf{X}_{n-2} \end{bmatrix} = \boldsymbol{\Phi} \begin{bmatrix} \hat{\theta}_0 & & & \\ & \hat{\theta}_1 & & \\ & & \ddots & \\ & & & \hat{\theta}_{n-1} \end{bmatrix} \boldsymbol{\Phi}^* \mathbf{X} = \boldsymbol{\Phi}(\hat{\theta} \circ \hat{\mathbf{X}})$$

$$w \quad * \quad h \quad \mathcal{F}^{-1} \quad \mathcal{F}(w) \quad \odot \quad \mathcal{F}(h)$$

https://towardsdatascience.com/deriving-convolution-from-first-principles-4ff124888028

$$w * h = \mathcal{F}^{-1}(\mathcal{F}(w) \odot \mathcal{F}(h))$$

$$\mathcal{F}(h) = \boldsymbol{\Phi}^{*} \boldsymbol{h}$$

What we have covered so far

- Discrete Fourier Transform (DFT) & Convolution theorem
  1. Convolution 연산은 circulant matrix로 표현 가능
  2. Circulant matrix는 교환 법칙 성립
  3. 모든 circulant matrices는 교환 법칙을 만족하기 때문에 동일한 eigenvectors를 가진다
  4. 모든 circulant matrices의 eigenvectors는 discrete fourier basis와 동일하다 ($C(w) = \boldsymbol{\Phi\Lambda\Phi^{*}}$) $\boldsymbol{\Phi}^{*} = \boldsymbol{\Phi}^{T}$

$$\boldsymbol{f}(\boldsymbol{X}) = \underbrace{\begin{bmatrix} b & c & & & a \\ a & b & c & & \\ & \ddots & \ddots & \ddots & \\ & & a & b & c \\ c & & & a & b \end{bmatrix}}_{C(w)} \underbrace{\begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_{n-1} \\ \mathbf{X}_{n-2} \end{bmatrix}}_{\mathbf{X}} = \boldsymbol{\Phi} \underbrace{\begin{bmatrix} \hat{\theta}_0 & & & \\ & \hat{\theta}_1 & & \\ & & \ddots & \\ & & & \hat{\theta}_{n-1} \end{bmatrix}}_{\boldsymbol{\Lambda}} \boldsymbol{\Phi}^{*} \mathbf{X} = \boldsymbol{\Phi}(\hat{\theta} \circ \hat{\mathbf{X}})$$

$$w \quad * \quad h \qquad \mathcal{F}^{-1} \quad \mathcal{F}(w) \quad \odot \quad \mathcal{F}(h)$$

https://petar-v.com/talks/GNN-Wednesday.pdf

$C(\theta) = C([b, c, 0, 0, \cdots, 0, a])$: a, b, c 를 weight로 하는 circulant matrix

$$w * h = \mathcal{F}^{-1}(\mathcal{F}(w) \odot \mathcal{F}(h))$$

What about graphs?

- **Graph Laplacian matrix**
  - Undirected graphs, L is:
    - Symmetric($L^T = L$)
    - Positive semi-definite($x^T Lx \geq 0 \; for \; all \; x \in \mathbb{R}^{|V|}$)

$$\Longrightarrow eigendecomposable! \; \mathbf{L} = \mathbf{\Phi\Lambda\Phi^*} \quad , \mathbf{\Phi\Phi^*} = \mathbf{I} \quad , \mathbf{\Phi} = [\emptyset_0, \emptyset_1, \cdots, \emptyset_{n-1}]$$

$\boldsymbol{Eigen - decomposition \; of \; graph \; Laplacian}$

Lap eigenvalues/Spectrum:

$$\underset{(n \times n)}{\mathbf{L} = \mathbf{\Phi}} \underset{\boldsymbol{\Lambda}(\boldsymbol{n} \times \boldsymbol{n})}{\begin{bmatrix} \hat{\theta}_0 & & & \\ & \hat{\theta}_1 & & \\ & & \ddots & \\ & & & \hat{\theta}_{n-1} \end{bmatrix}} \mathbf{\Phi}^*$$

Lap eigenvectors/
Fourier functions (orthonormal basis)

Unnomalized Laplacian $\quad \mathbf{L} = \mathbf{D} - \mathbf{A}$

Nomalized Laplacian

$$\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{A})\mathbf{D}^{-\frac{1}{2}}$$
$$= \mathbf{D}^{-\frac{1}{2}}(\mathbf{D}^{\frac{1}{2}} - \mathbf{A}\mathbf{D}^{-\frac{1}{2}})$$
$$= \mathbf{D}^{-\frac{1}{2}}(\mathbf{D}^{\frac{1}{2}} - \mathbf{A}\mathbf{D}^{-\frac{1}{2}})$$
$$= \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$$

Adjacency(Laplaciain) matrix가 graph shift matrix & eigenvector가 graph fourier basis인 이유 증명: Gavili, A., & Zhang, X. P. (2017). On the shift operator, graph frequency, and optimal filtering in graph signal processing

What about graphs?

$$w * h = \mathcal{F}^{-1}(\mathcal{F}(w) \odot \mathcal{F}(h))$$

Filter(모르는 값) 아는 값 아는 값 모르는 값 아는 값

Eigen – decomposition of graph Laplacian

General graph

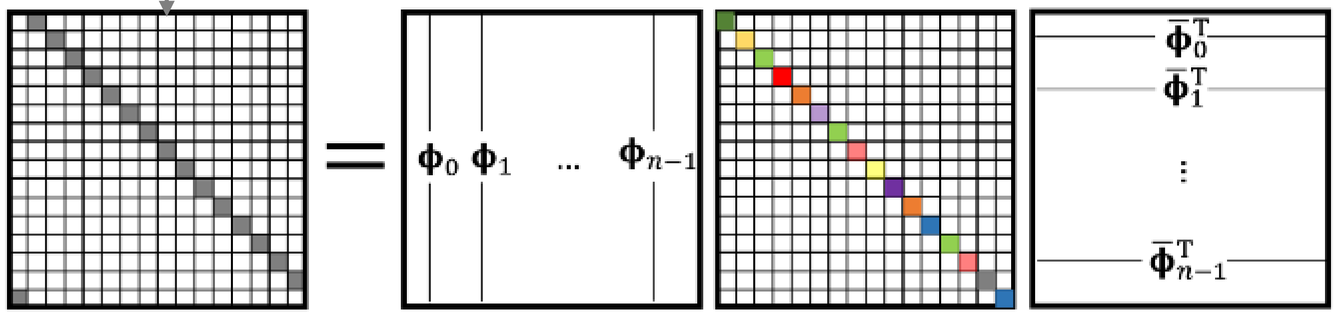
Lap eigenvalues/Spectrum:

Graph shift matrix

$$\mathbf{L} = \underset{(n \times n)}{\boldsymbol{\Phi}} \underset{\boldsymbol{\Lambda}(n \times n)}{\begin{bmatrix} \hat{\theta}_0 & & & \\ & \hat{\theta}_1 & & \\ & & \ddots & \\ & & & \hat{\theta}_{n-1} \end{bmatrix}} \underset{(n \times n)}{\boldsymbol{\Phi}^*}$$

유사한 역할 수행

Circulants



$\mathbf{S}^T$ Shift matrix $\boldsymbol{\Phi}$ $\boldsymbol{\Lambda}$ $\boldsymbol{\Phi}^*$

Q. laplacian matirx를 고유값분해하면 이미 고유값, 고유벡터를 다 구할 수 있는데 고유값을 학습 파라미터로 사용한다는게 이해가 잘 안가서요 ..

Lap eigenvalues

$$\mathbf{L} * h = \boldsymbol{\Phi} \; \boldsymbol{\Lambda} \; \boldsymbol{\Phi}^* \; \mathbf{X}$$

Page 3 내용과 유사하게 적용됨

실제 filter(w)를 통해 학습할 때

Lap eigenvectors

$$w * h = \boldsymbol{\Phi} \; \boldsymbol{\Lambda} \; \boldsymbol{\Phi}^* \; \mathbf{X}$$ Node signal

Filter(모르는 값) 아는 값 아는 값 모르는 값 아는 값

Filter 대신 학습할 값
(Lap eigenvalues X)

A:
w를 직접 학습시키는 것이 아니라 고유값 분해해서 eigenvalues 에 해당하는 부분을 학습시킴
이 때, $\boldsymbol{\Phi}$, $\boldsymbol{\Phi}^*$는 graph laplacian이 shift matrix 역할을 하므로 laplacian의 eigenvectors 로 사용할 수 있음. 그러나, $\boldsymbol{\Lambda}$ 는 모름.
따라서, filter 를 대신 학습할 parameters 해당됨

Graph Convolution

- Spectral convolution

$\mathcal{F}(\mathrm{X}) = \boldsymbol{\Phi}^{*}\mathrm{X}$: fourier transform

$\odot$: *Pointwise product*

$$\underset{(\boldsymbol{n}\times\mathbf{1})}{\widehat{w}} = \begin{bmatrix} \widehat{w}(\lambda_0) \\ \vdots \\ \widehat{w}(\lambda_{n-1}) \end{bmatrix}$$

$$\underset{(\boldsymbol{n}\times\boldsymbol{n})}{\widehat{w(\boldsymbol{\Lambda})}} = diag(\widehat{w}) = \begin{bmatrix} \widehat{w}(\lambda_0) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \widehat{w}(\lambda_{n-1}) \end{bmatrix}$$

signal h on graph — *Graph Fourier transform*

$$\underset{filter/kernel}{w} * h = \mathcal{F}^{-1}\left(\mathcal{F}(w) \odot \mathcal{F}(h)\right)$$

$$\underset{(\boldsymbol{n}\times\boldsymbol{n})}{\boldsymbol{\Phi}} \qquad \underset{(\boldsymbol{n}\times\boldsymbol{n})(\boldsymbol{n}\times\mathbf{1})}{\boldsymbol{\Phi}^{*}\boldsymbol{w} = \widehat{\boldsymbol{w}}} \qquad \underset{(\boldsymbol{n}\times\boldsymbol{n})\ (\boldsymbol{n}\times\mathbf{1})}{\boldsymbol{\Phi}^{*}\boldsymbol{h}}$$

learning parameterss/filter

$$= \underset{(\boldsymbol{n}\times\boldsymbol{n})}{\boldsymbol{\Phi}}(\underset{(\boldsymbol{n}\times\mathbf{1})}{\widehat{\boldsymbol{w}}} \odot \underset{(\boldsymbol{n}\times\mathbf{1})}{\boldsymbol{\Phi}^{*}\boldsymbol{h}})$$

$$= \underset{(\boldsymbol{n}\times\boldsymbol{n})}{\boldsymbol{\Phi}}\ \underset{(\boldsymbol{n}\times\boldsymbol{n})}{\widehat{\boldsymbol{w}}(\boldsymbol{\Lambda})}\ \underset{(\boldsymbol{n}\times\mathbf{1})}{\boldsymbol{\Phi}^{*}\boldsymbol{h}}$$