목차

$\mathbb{R}^2$

- convolution
- fourier transform
- laplacian matrix

$\rightarrow$

- graph fourier
- graph convolution

+ "chebyshev polynomial"

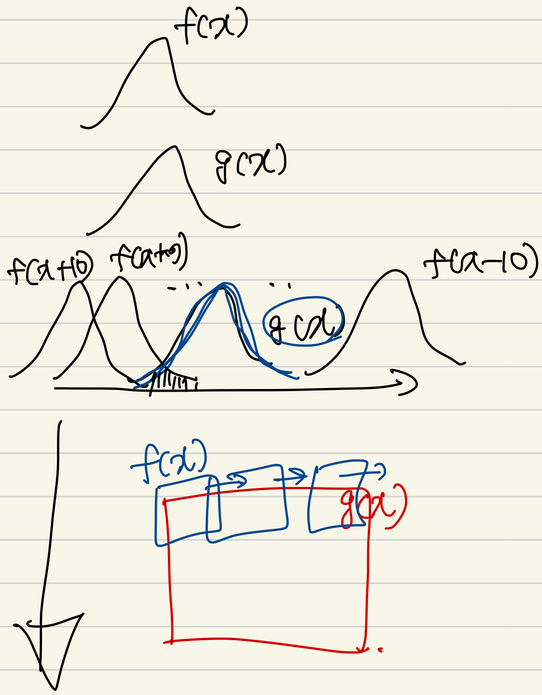
논문!

Convolution 이란?

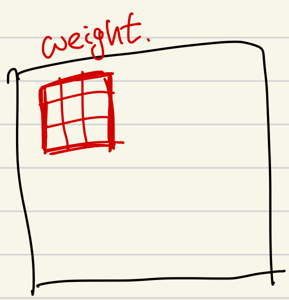
함수 f, g : ℝ → ℝ 일 때

$$(f*g)(x) := \int_{-\infty}^{\infty} f(x-t)\,g(t)\,dt = \int_{-\infty}^{\infty} g(x-t)\,f(t)\,dt$$

$\boxed{\text{convolution 기호}}$



= g(x)가 옮겨가는거.



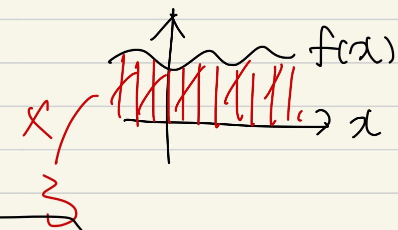
* conv filter라는 작은 dimension으로 낮추면 연산량↓

Fourier Transform

F: fourier transform

$f, g : \mathbb{R} \to \mathbb{R}$

x, ξ (크시) $\in \mathbb{R}$



정의 $F[f(x)](\xi) := \boxed{\int_{-\infty}^{\infty} f(x) \cdot e^{-2\pi i x \xi} \, dx} = F(\xi)$

정의2 $F^{-1}[f](x) := F[f](-x)$

ex $\int_{-\infty}^{\infty} f(-x) \, e^{2\pi i \xi}$

성질

① $F(F^{-1}(f)) = f$

② $F(f * g) = F(f) \cdot F(g)$

convolution

$f(x)$

$\downarrow$

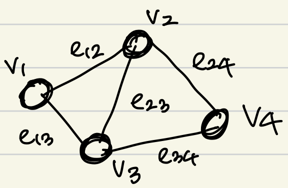
Fourier

$\downarrow$

$f'(\xi)$

Laplacian Matrix

undirected graph = (vertex, edge)



$\Longrightarrow$

graph laplacian

[Σrow   Σrow]

$L_G =$ Degree Matrix $-$ Adjacency Matrix

노드에 간선이 몇개 연결되어 있는지.

연결간선.

Σrow=0

[노드개수 : N
간선개수 : M]

$$N\left\{\begin{bmatrix} 2 & & & \\ & 3 & & \\ & & 3 & \\ & & & 2 \end{bmatrix}\right. \qquad \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 & 1 \\ 3 & 1 & 1 & 0 & 1 \\ 4 & 0 & 1 & 1 & 0 \end{array}$$

특이점

1) 대칭이다. (adjacency matrix가 대칭이니까)

2) $\boxed{B^T B = L}$인 (M×N) matrix B가 존재한다.

"graph theory"

$(N\cdot M)\,(M\cdot N) = (N\cdot N)$

B ⇒ indicicous matrix

<L_G의 성질을 파악해보자>

① 임의의 $x \in \mathbb{R}^N$에서,

$$\underbrace{X^T\cdot L_G\cdot X}_{} \overset{(2)}{=} \underbrace{X^T}_{(1,n)}\cdot\underbrace{(B^T}_{(n\cdot m)}\underbrace{B)}_{(m\cdot n)}\cdot\underbrace{X}_{(n,1)} \qquad \mathbb{R}^m$$

$$= \|BX\|_2^2 \ge 0 \text{ 이므로,}$$

왜??

$(BX)^T\cdot(BX) = \|BX\|_2^2$

$y^T y = \|y\|_2^2 \qquad y = (y_1, \cdots y_N)$

$= y_1^2 + \cdots y_N^2$

positive-semi definite 의 정의는 모든 X에 대해 $X^T\cdot A\cdot X \ge 0$ 인것이다.

→ 이면 모든 eigen value ≥ 0.

$\boxed{\lambda_1, \cdots \lambda_N \ge 0}$

(N: vertex의 수)

$V_1 := \left(\frac{1}{\sqrt{N}}, \cdots, \frac{1}{\sqrt{N}}\right)$이면

N이 있어서 $\|\vec{V_1}\|_2 = 1$이다.

eigenvalue=0 → eigenvector = (1, 1, ⋯ 1) 이 되겠구나.

laplacian : $\begin{bmatrix} a & -a \\ -b & b \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0 \cdot \vec{v_1}$

eigen value　eigen vector

$(\overbrace{1, 1, \cdots 1}^{N})$

$0 = \lambda_1 \le \cdots \le \lambda_n$ 이므로

0은 가장 작다고 말할 수 있다.

(결국 $\Sigma row = 0$ 이라서 eigenvalue(λ_1)=0 이 된다)

2 <선형대수의 성질>

A가 실수이고 대칭 ⟷ (orthonormal) eigenvalue가 실수 eigenvector의 set $\{v_1, \cdots, v_N\}$은 직교하고 선형독립.

기억하고 있자! eigen-decomposition

정방행렬이니까 eigen decomposition 가능

3 $L_G = Q \Lambda Q^T$

$$\Lambda = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{bmatrix}$$

$$Q = [v_1, \cdots, v_N]$$

함수 input, 함수 output

Graph Fourier

임의의 함수   그래프 위에서의 함수의 정의.

graph signal $f : \text{Vertex} \to \mathbb{R}$

$F = (f(\text{vertex }1), \cdots f(\text{vertex }N)) \in \mathbb{R}^N$

fourier transform of signal F $\text{fourier}(F) := Q^T F$ $(Q = [v_1, \cdots, v_N]$ $(N \times N)$

Laplacian matrix의 eigen-decomposition $Q \Lambda Q^T$ eigenvector

(Q) 앞에서 정의한 fourier랑 어떻게 같은가?

앞: euclidian space

지금: non-euclidian space.

"fourier series" ← fourier transform의 special case

= fourier basis의 linear combination

⇓

eigenvector

$$\text{fourier}(F) = \begin{bmatrix} - v_1^T - \\ \vdots \\ - v_N^T - \end{bmatrix} \begin{bmatrix} f(\text{vertex }1) \\ \vdots \\ f(\text{vertex }N) \end{bmatrix}$$

F

$$= \begin{bmatrix} v_1^T F \\ \vdots \\ v_N^T F \end{bmatrix}$$

eigenvalue로 표현가능.

: $\{\lambda_1, \cdots, \lambda_N\}$ spectrum := frequency

$\mathbb{R}^N \longrightarrow \mathbb{R}^N$

inverse는 어떻게 될까? 계산해보자.

$(\text{fourier}(F) := Q^T F)$이고,

$F = \text{fourier}^{-1}(\underbrace{Q^T F}_{g})$ 인데 $\quad Q^T F = g$ 라고 하면

$Q^T = Q^{-1}$인 성질때문에

$Q \cdot Q^T F = Q \cdot g$

따라서, $\text{fourier}^{-1}(g) = Qg$

왜 $Q^T = Q^{-1}$??

$$Q = \begin{bmatrix} | & & | \\ e_1 & \cdots & e_m \\ | & & | \end{bmatrix}, \quad Q^T = \begin{bmatrix} - & e_1^T & - \\ & \vdots & \\ - & e_m^T & - \end{bmatrix}$$

$e_1^2 \quad e_1 e_2 \cdots e_1 e_m$

$$Q^T \cdot Q = \begin{bmatrix} - e_1^T - \\ - e_m^T - \end{bmatrix} \begin{bmatrix} | & & | \\ e_1 & \cdots & e_m \\ | & & | \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} \overset{1}{e_1^T e_1} & \overset{0}{e_1^T e_2} & \cdots & \overset{0}{e_1^T e_m} \\ & \ddots & & \\ & & & \underset{1}{e_m^T e_m} \end{bmatrix}} = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix} = I$$

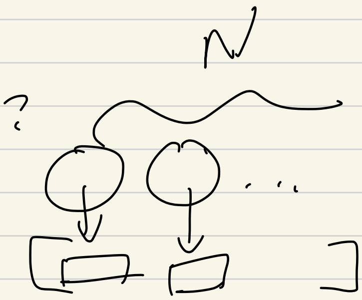
$(e_1^T e_1 = \|e_1\|_2^2$ 이니까!$)$

$$Q^T = Q^{-1}$$

Graph Convolution (spectral convolution)

기존의 convolution ⇒ $f*g(x) = \int_{-\infty}^{\infty} f(x-t)\,g(t)\,dt$

(Q) 그렇다면 그래프에서 convolution은 어떻게 정의할까?



시그널 $x, y \in \mathbb{R}^N$ 에 대해서

$$x * y := \text{fourier}^{-1}\big(\underbrace{\text{fourier}(x)}_{\mathbb{R}^N} \cdot \underbrace{\text{fourier}(y)}_{\mathbb{R}^N}\big)$$

(★ 정의 ★)

$$= Q\left(\left[Q^T x\right]\right) \cdot (Q^T y)$$

원소곱하려고 만든 방법.

$v_1 = (1, 2, 3)$

$v_2 = (2, 4, 6)$

$$Q^T x = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$$Q^T y = \begin{bmatrix} d \\ e \\ f \end{bmatrix}$$

$$= Q\left(\begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \cdot \begin{bmatrix} d \\ e \\ f \end{bmatrix}\right)$$

$$= Q \cdot \left(\begin{bmatrix} ad \\ be \\ cf \end{bmatrix}\right).$$

1) graph fourier
2) graph convolution

저자는 아무것도 만들지 않음.

Spectral Convolution

→ first-order approximation

→ 그래프 각각 → computation power ↑

"흩어져 있던걸 모아왔다"

→ 수학을 딥러닝으로 끌어왔다.

spatial computation → power ⇓

(인접노드만)

* semi-supervised ⇒ regularization term 더하지 않아도 된다.

논문

normalized graph laplacian (수학)

$$\tilde{L}_G = D^{-\frac{1}{2}} \underbrace{L_G}_{D-A} D^{-\frac{1}{2}} = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \Rightarrow D^{-\frac{1}{2}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2^{-\frac{1}{2}} & 0 \\ 0 & 0 & 3^{-\frac{1}{2}} \end{bmatrix}$$

diagonal일때만.

$$\begin{matrix} a & & \\ & b & \\ & & c \end{matrix} \rightarrow a^{-\frac{1}{2}}$$

$$0 \le \lambda_1 \le \cdots \lambda_n \le ??$$

$$0 \le \tilde{\lambda}_1 \le \cdots \tilde{\lambda}_n \le 2 \; *$$

Convolution Filter

러닝할 파라미터

$$\theta = (\theta_1, \cdots, \theta_N) \in \mathbb{R}^N$$

함수 g_θ : $\overset{\text{input}}{(\tilde{\lambda}_i)} \to (\theta_i)$이고

$g_\theta(\tilde{\lambda}_i) = \theta_i$ 라고 하면

$\Lambda = \begin{bmatrix} \tilde{\lambda}_1 & & 0 \\ & \ddots & \\ 0 & & \tilde{\lambda}_N \end{bmatrix}$ 이므로, $\quad \tilde{Q} = \begin{bmatrix} | & & | \\ \tilde{v}_1 & \cdots & \tilde{v}_N \\ | & & | \end{bmatrix}$

$g_\theta(\Lambda) = \begin{bmatrix} \theta_1 & & 0 \\ & \ddots & \\ 0 & & \theta_N \end{bmatrix}$ 이다.

g_θ를 convolution filter라고 부르기로 함!

$G_\theta := \text{fourier}^{-1}(g_\theta(\Lambda))$ 라고 정의하면,

$= \tilde{Q} \cdot \theta = (n \times n) \times (n \times 1) \Rightarrow (n \times 1)$

signal x의 spectral convolution (filter $g_\theta = g_\theta(\Lambda)$)은

$g_\theta * x := G_\theta * x$ 로 정의한다.

$\boxed{x * y} := \text{fourier}^{-1}(\overset{\mathbb{R}}{\text{fourier}(x)} \cdot \overset{\mathbb{R}^N}{\text{fourier}(y)})$ (★ 정의 ★)

$= \boxed{\tilde{Q}\,([\tilde{Q}^T x]) \cdot (\tilde{Q}^T y)}$

$$= \text{fourier}^{-1}(\boxed{g_\theta(\Lambda)}) \overset{\text{conv}}{*} x$$

$$= \tilde{Q} \cdot \text{diag}(\tilde{Q}^T \cdot G_\theta) \cdot \tilde{Q}^T \cdot x$$

$G_\theta = \tilde{Q}\theta$

$[\cancel{\tilde{Q}^T} \cdot \cancel{\tilde{Q}} \cdot \theta]$

I_N

$N \times N$

$$= \tilde{Q} \cdot \text{diag}(\theta) \cdot \boxed{\tilde{Q}^T} \cdot \boxed{x}$$

$$= \tilde{Q} \cdot g_\theta(\Lambda) \cdot \tilde{Q}^T \cdot x$$

$\hat{g}$ 구하기 어렵다 ⇒ approximation : spectral ⇒ spatial

Chebyshev Polynomial

$$\begin{cases} T_0(x)=1,\ T_1(x)=x,\ T_2(x)=2x^2-1,\ T_3(x)=\cdots \\ T_{n+1}(x) = 2x\cdot T_n(x) - T_{n-1}(x) \end{cases}$$

x가 $-1 \sim 1$일 때, + $f(x)$가 smooth, continuous이면

$$\boxed{f(x)= a_0T_0(x)+a_1T_1(x)+a_2T_2(x)+\cdots}$$

인 a들이 존재한다.

그런데 normalized graph laplacian은,

아까 $0=\tilde{\lambda}_0 \le \cdots \le \tilde{\lambda}_n \le 2$ 라고 해서,

새로운 x의 범위.

$$-1 \le \boxed{\frac{2}{\tilde{\lambda}_n}\cdot\tilde{\lambda}_i - 1} \le 1$$ 범위로

바꿀 수 있다.

→ chebyshev polynomial을 이용하면,

$$g_\theta(\Lambda)=\sum_{i=0}^{K} a_i \cdot T_i\left(\boxed{\left(\frac{2}{\tilde{\lambda}_n}\right)\Lambda - I_N}\right)$$ 꼴로

바꿀 수 있다.

(이건 논문 내용임, wavelets on graphs (2012))

그러므로, 우리가 구하고 싶어하는

$\tilde{Q}\, \boxed{g_\theta(\Lambda)}\, \tilde{Q}^T$ $\qquad a_0 \quad a_1$

$$= \tilde{Q}\left(\sum_{i=0}^{K} a_i \cdot T_i\left(\frac{2}{\tilde{\lambda}_n}\Lambda - I_n\right) \right)\tilde{Q}^T$$

$$= \tilde{Q}\left(a_0 \cdot \underbrace{\boxed{T_0\left(\frac{2}{\tilde{\lambda}_n}\Lambda - I_n\right)}}_{=I_N}\right)\cdot \tilde{Q}^T = a_0 \cdot I_N$$

$$+ \tilde{Q}\left(a_1 \cdot \underbrace{\boxed{T_1\left(\frac{2}{\tilde{\lambda}_n}\Lambda - I_n\right)}}_{=\left(\frac{2}{\tilde{\lambda}_n}\Lambda - I_n\right)}\right)\cdot \tilde{Q}^T + \cdots +$$

$$\left(a_1 \cdot \frac{2}{\tilde{\lambda}_n}\underbrace{\boxed{\tilde{Q}\cdot\Lambda\cdot\tilde{Q}^T}}_{=\tilde{L}_G}\right) - a_1 \cancel{\tilde{Q}\cdot\tilde{Q}^T}\cdot I_n$$

$$= a_1 \cdot \underbrace{\left(\frac{2}{\tilde{\lambda}_n}\cdot\tilde{L}_G - I_n\right)}$$

$\bar{L}_G := \frac{2}{\tilde{\lambda}_n}\cdot\tilde{L}_G - I_n$ 라고 정의한다.

$\Rightarrow$ first-order approximation을 써보자! $(K=1)$

$$g_\theta * x \approx a_0 x + a_1 \bar{L}_G x$$

$$= a_0 x - a_1 (D^{-\frac{1}{2}} A D^{-\frac{1}{2}})x$$

$\left(\tilde{L}_G = D^{-\frac{1}{2}} \underset{D-A}{L_G} D^{-\frac{1}{2}} = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right)$이므로!

$a_0 = -a_1 = b$ 라고 가정한다.

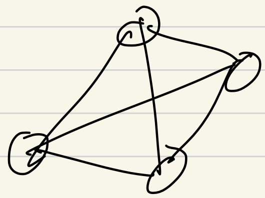
$$g_\theta * x \approx b\left(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right)x$$

$$\begin{cases} \underbrace{\tilde{A} := A + I_N} \\ \tilde{D} := D + I_N \end{cases}$$ 이라고 해서

$$g_\theta * x \approx b\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}\right)x.$$

목적 ⇒ b를 train 하자.

↓ layer 적용으로 확장.



$\sigma(WX + b) \rightarrow$ linear

$\sigma\left(\int f(x)g(x)\right) \rightarrow$ conv.

실제 shape을 살펴보자.

$$g_\theta * x \approx b(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}})x.$$

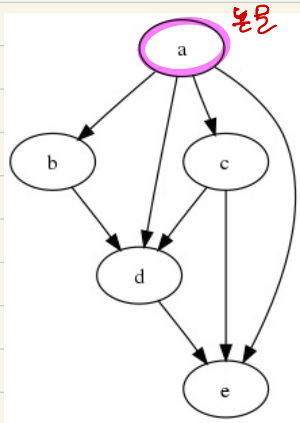
목적 ⇒ b를 train하자.

$$\begin{pmatrix} \tilde{D} = D + I_N \\ \tilde{A} = A + I_N \end{pmatrix}$$

N: node
C: feature

*논문 p.6에서 un

"citation network"



$$D = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix} + I$$

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix} + I$$

Input layer

$$X = [x_1, \cdots x_C] \in \mathbb{R}^{N \times C}$$

$$\theta \in \mathbb{R}^{C \times F}$$ 필터수

$$Z := \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta$$

↓ conv layer.
↓ activation

$$X = \begin{array}{c|cccc} & class1 & & \cdots & classC \\ \hline 논문1 & 1 & 0 & 0 \cdots & 0 \\ 논문2 & 2 & 0 & 0 & 1 \\ \vdots & & & & \\ 논문N & & & & \end{array}$$

$$\Theta = \begin{array}{c|ccc} & f1 & \cdots & fN \\ \hline class1 & & & \\ \vdots & & & \\ classC & & & \end{array}$$