Graph Fourier Transform

游承軒

2022/01/18

Outline

- Introduction
    - Graph Signal
    - Graph Shift
    - Graph Filter
    - Graph Fourier Transform (GFT)
    - Properties and Example
- Application
    - Image Coding using GFT
    - Graph Convolutional Network (GCN)
- Conclusion
- Reference

Outline

- Introduction
  - Graph Signal
  - Graph Shift
  - Graph Filter
  - Graph Fourier Transform (GFT)
  - Properties and Example
- Application
  - Image Coding using GFT
  - Graph Convolutional Network (GCN)
- Conclusion
- Reference

Introduction

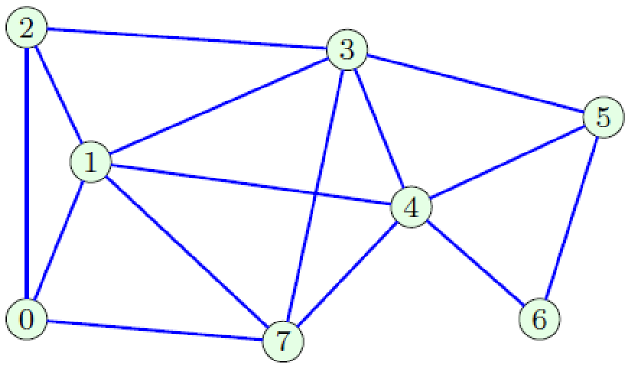
Graph: G=(V, E), N=|V|

$A \in \mathbb{R}^{N \times N}$: adjacency matrix, $A_{ij} = \begin{cases} 0 & if \;\; e_{i,j} \notin E \\ w(i,j) & otherwise \end{cases}$

$\mathrm{D} \in \mathbb{R}^{N \times N}$: degree matrix, $D_{i,j} = \begin{cases} \sum_j A_{ij} & if \;\; i = j \\ 0 & otherwise \end{cases}$

$L \in \mathbb{R}^{N \times N}$: Laplacian matrix (graph Laplacian), L = D - A

$L^{sym} \in \mathbb{R}^{N \times N}$: normalized Laplacian matrix, $L^{sym} = D^{-1/2} L D^{-1/2}$



undirected graph

Introduction

Discrete Fourier Transform: $$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi nk}{N}}$$

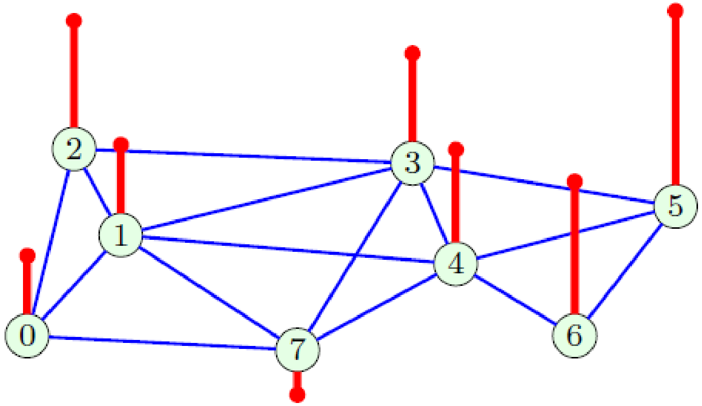
$$x[n] = \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi nk}{N}}$$

How to connect them?

Graph Signal

Signals on graphs(vertex domain signals):

$$\mathbf{x} = [x(0),\ x(1),\ \ldots,\ x(N-1)]^T$$

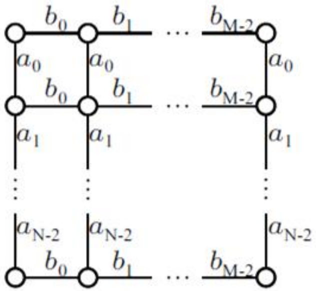


Graph Signal

Several common signals can be transformed into graph signals

(1) time series
(2) digital image
(3) weather stations

(a) Time series

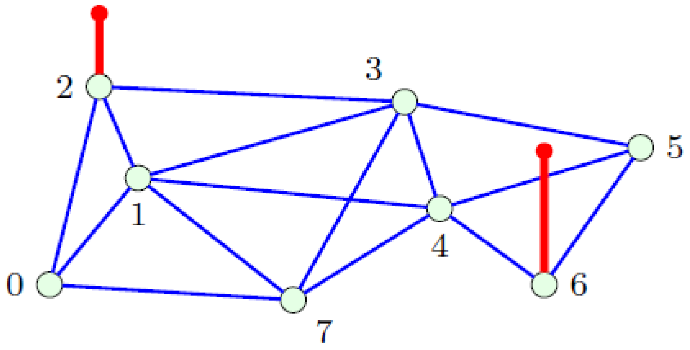


(b) Digital image

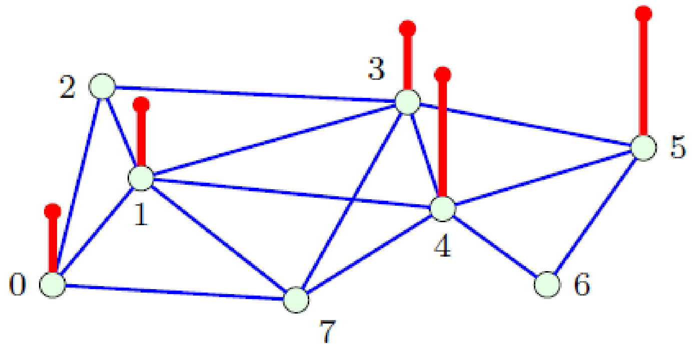
(c) Weather stations across the U.S.

Graph Shift with A

A graph shift is the movement of the signal sample from the vertex n along all walks, with the length equal to one.



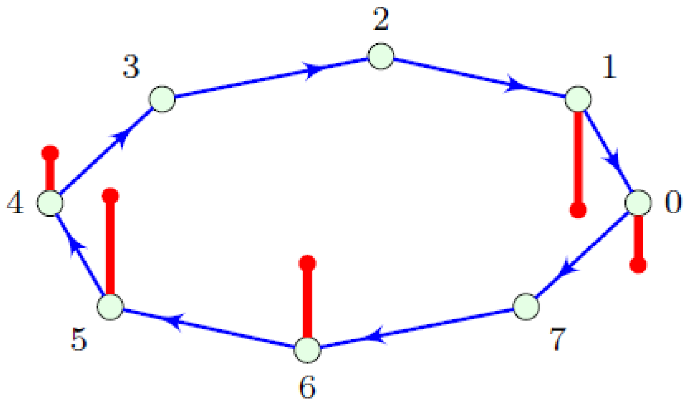
$\mathbf{x}$



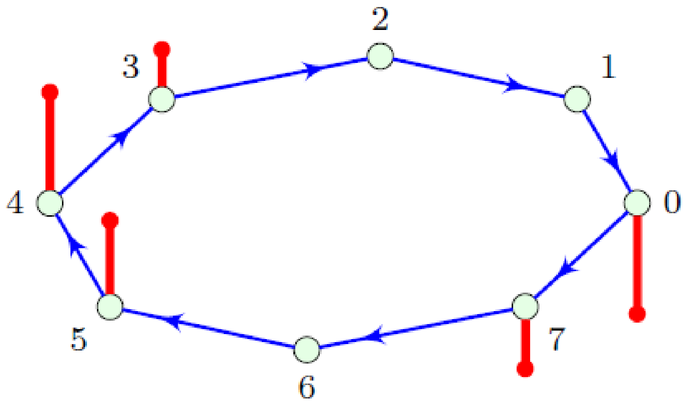
$\mathbf{x}_1 = \mathbf{A}\mathbf{x}$

Graph Shift with A

Considering a graph shift on a directed circular graph, it is like signals shifted by 1 in traditional DSP



$\mathbf{x}$



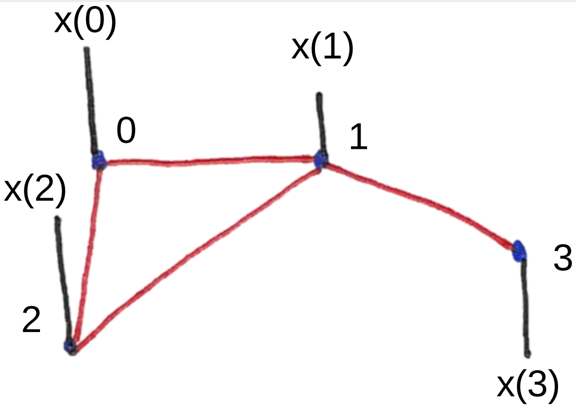
$\mathbf{x}_1 = \mathbf{A}\mathbf{x}$

Graph Shift with A

In general, a graph signal shifted by m is obtained as a shift by 1 of the graph signal shifted by m-1:

$$\mathrm{x}_m = \mathbf{A}\mathrm{x}_{m-1} = \mathbf{A}^m \mathrm{x}$$

Graph Shift with L



$$x_1 = \mathbf{L}x = \mathbf{D}x - \mathbf{A}x$$

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, x = \begin{bmatrix} 4 \\ 2 \\ 4 \\ -3 \end{bmatrix}$$

Lx : # of neighbors*signal on it– sum of signals over its neighbors

= Sum of signal differences between neighbors

= $\sum_{j \in N(i)} (x(i) - x(j))$

- For node 0, $x_1(0)$ = 2*4-2-4 = 1*4-2+1*4-4
- In practice, we often use L or normalized L for graph shift for better mathematical properties than A

Graph Filter

The output signal from a system on a graph can be written as

$$
\begin{aligned}
\mathrm{y} &= h_0 \mathbf{L}^0 \mathrm{x} + h_1 \mathbf{L}^1 \mathrm{x} + \ldots + h_{M-1} \mathbf{L}^{M-1} \mathrm{x} \\
&= H(\mathbf{L}) \mathrm{x}
\end{aligned}
$$

H(L): graph filter, a matrix polynomial of L

Graph Fourier Transform

Spectral decomposition of L: $L = U\Lambda U^{-1}$

U: columns are eigenvectors of L

Λ: diagonal matrix, diagonal entries are eigenvalues of L

$$\mathrm{y} = h_0 \mathbf{U}\mathbf{\Lambda}^0 \mathbf{U}^{-1}\mathrm{x} + h_1 \mathbf{U}\mathbf{\Lambda}^1 \mathbf{U}^{-1}\mathrm{x} + \ldots + h_{M-1} \mathbf{U}\mathbf{\Lambda}^{M-1} \mathbf{U}^{-1}\mathrm{x}$$
$$= \mathbf{U} H(\mathbf{\Lambda}) \mathbf{U}^{-1}\mathrm{x}$$

Graph Fourier Transform

Multiplied by U^{-1} => $\mathbf{U}^{-1}\mathrm{y} = H(\mathbf{\Lambda})\mathbf{U}^{-1}\mathrm{x}$

U^{-1} : graph fourier transform(GFT) matrix

- GFT: $X = U^{-1}x$
- Inverse GFT: $x = UX$

$\mathrm{Y}=H(\mathbf{\Lambda})\mathrm{X}$

X, Y : spectral domain graph signals

$H(\mathbf{\Lambda})$: filter in spectral domain

Graph Fourier Transform

$$X(l) = \sum_{i=0}^{N-1} x(i) u_l(i)$$ orthogonal basis

=> GFT is a projection on the eigenspace of the graph shift operator

Comparison with DFT

- GFT

$$X(l) = \sum_{i=0}^{N-1} x(i) u_l(i)$$

$$x(i) = \sum_{l=0}^{N-1} X(l) u_l(i)$$

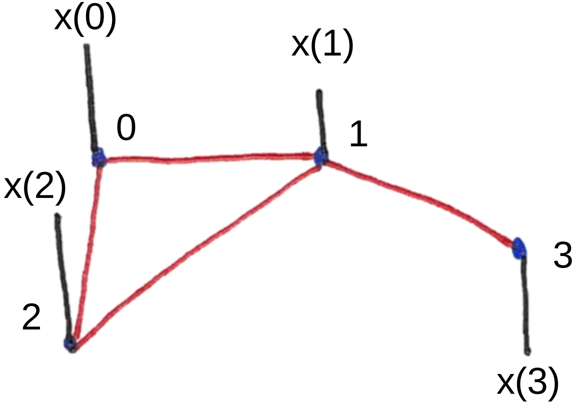
- DFT

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi nk}{N}}$$

$$x[n] = \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi nk}{N}}$$

Example of GFT

$$U = \begin{array}{c} \begin{bmatrix} -0.5 & 0.4082 & 0.7071 & -0.2887 \\ -0.5 & 0 & 0 & 0.8660 \\ -0.5 & 0.4082 & -0.7071 & -0.2887 \\ -0.5 & -0.8165 & 0 & -0.2887 \end{bmatrix} \\ \begin{array}{llll} u_1 & & & \end{array} \end{array} \qquad \Lambda = \begin{array}{c} \begin{array}{llll} \lambda_1 & & & \end{array} \\ \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 3 & \\ & & & 4 \end{bmatrix} \end{array}$$



$$x = \begin{bmatrix} 4 \\ 2 \\ 4 \\ -3 \end{bmatrix} \qquad X = \begin{bmatrix} -3.5 \\ 5.7155 \\ 0 \\ 0.2887 \end{bmatrix}$$

Frequency of graph signals

- frequency ↑ ⇨ signal variation ↑
- Graph shift with L (weighted): $x_1(i) = Lx(i) = \sum_{j \in V} w_{i,j}(x(i) - x(j))$

$$x^T Lx = \sum_{i \in V} x(i) \sum_{j \in V} w_{i,j}(x(i) - x(j))$$

$$= \sum_{i \in V} \sum_{j \in V} w_{i,j}(x^2(i) - x(i)x(j))$$

$$= \frac{1}{2} \sum_{i \in V} \sum_{j \in V} w_{i,j}(x^2(i) - x(i)x(j) + x^2(j) - x(i)x(j))$$

$$= \frac{1}{2} \sum_{i \in V} \sum_{j \in V} w_{i,j}(x(i) - x(j))^2$$

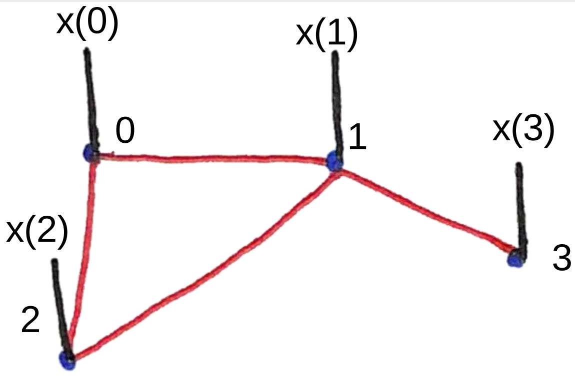
=>a measure of signal variation
(frequency)

Frequency of graph signals

- Consider eigenvector u_l => $u_l{}^T L u_l = \lambda_l u_l{}^T u_l = \lambda_l$
- $\lambda_l \uparrow \;\; \Rightarrow$ frequency of $u_l \uparrow$
- Inner product with $u_l \uparrow \;\; \Rightarrow X(l) \uparrow$ $\quad \left(X(l) = \sum_{i=0}^{N-1} x(i) u_l(i) \right)$
- If X(l) of the signal is large, it contains a lot of component u_l

Similar to DSP, signal values in spectral domain reflects frequency components of the signal

Example of GFT



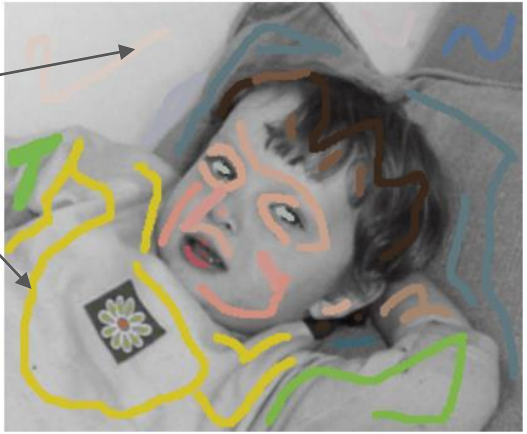
$$x = \begin{bmatrix} 4 \\ 4 \\ 4 \\ 4 \end{bmatrix} \qquad X = \begin{bmatrix} -8 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Outline

- Introduction
  - Graph Signal
  - Graph Shift
  - Graph Filter
  - Graph Fourier Transform (GFT)
  - Properties and Example
- Application
  - Image Coding using GFT
  - Graph Convolutional Network (GCN)
- Conclusion
- Reference

Colorization

representative pixels (RPs) x



luminance image y

$$u = Cx$$

chrominance image u

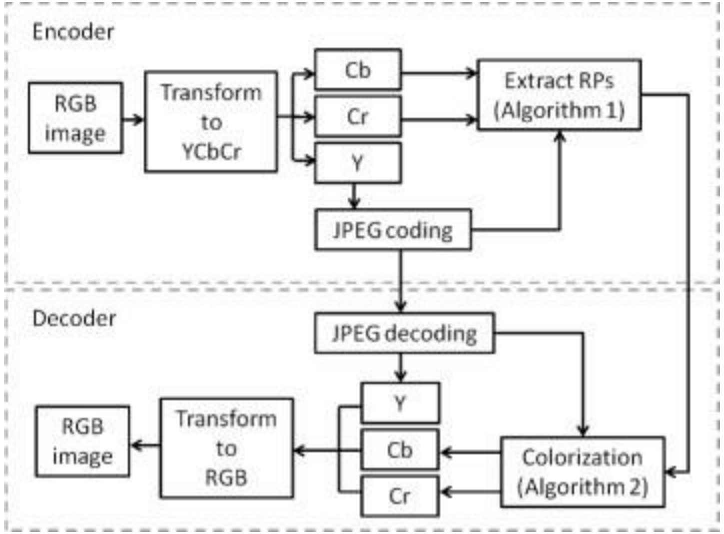
$$u = Cx$$

Colorization Based Coding

- C = (I - B)$^{-1}$ $\qquad B_{ij} = \begin{cases} 0 & if\ \ i\ is\ in\ \mathrm{RP}\ or\ j \notin 8-neighboring\ pixels\ of\ \ i \\ \dfrac{1}{S_i}\exp\left(\dfrac{-(y_i - y_j)^2}{2\sigma_i^{\ 2}}\right) & \end{cases}$

- Colorization-based image coding
- Recovering error $\min_x \| u^* - Cx \|_2^2 \ \ s.t.\ x_i = 0\ if\ \ i\ is\ not\ in\ \mathrm{RP}$

u*: original chrominance image
x: RP vector $\in \mathbb{R}^{\mathrm{MN}}$
Cx: recovered chrominance image



Colorization-based image coding

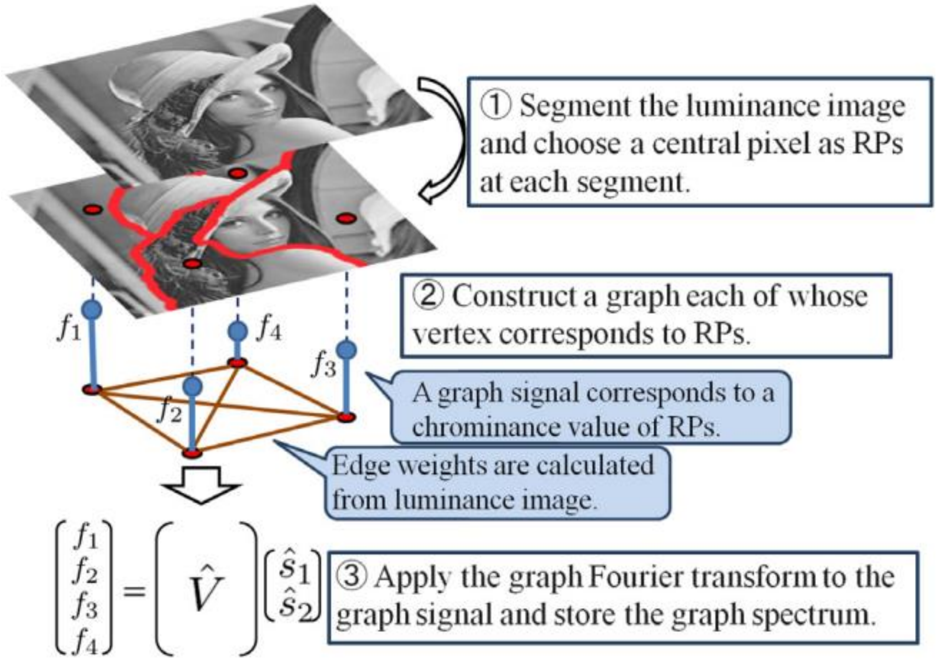
Colorization Based Coding Using GFT

- Graph construction of p RPs

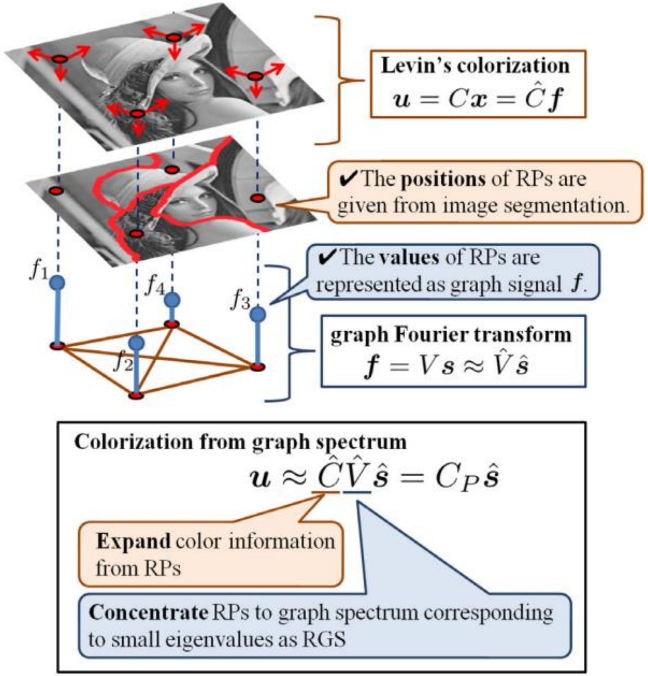
$$w_{i,j} = \exp(-\alpha \Delta d_{(i,j)}) \exp(-\beta \Delta y_{(i,j)})$$

- $\Delta d_{(i,j)}$: distance between RPs(vertices)
- $\Delta y_{(i,j)}$: difference of the luminance values between vertices

Colorization Based Coding Using GFT



Colorization Based Coding Using GFT



$u = Cx = \hat{C}f$ $\quad$ $f \in \mathbb{R}^P,\ C \in \mathbb{R}^{MN\times P}$

$f = Vs \approx \hat{V}\hat{s}$ $\quad$ $\hat{V} \in \mathbb{R}^{P\times c}$

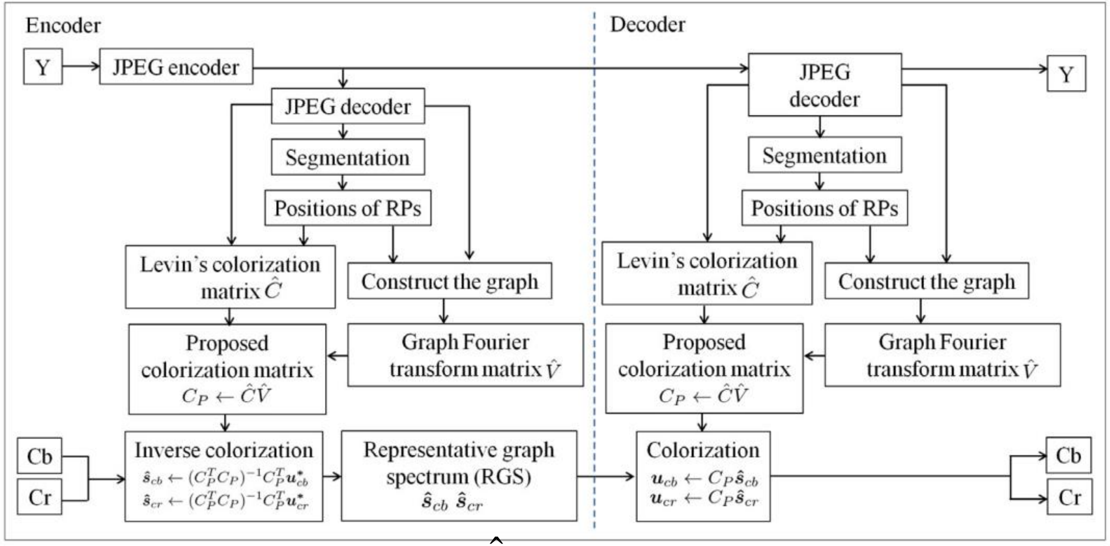
$u \approx \hat{C}\hat{V}\hat{s} = C_p\hat{s}$ $\quad$ $\hat{s} \in \mathbb{R}^c$

Recovering error: $\min_s \| u^* - C_p\hat{s} \|_2^2$

$$\hat{s} = (C_p{}^T C_p)^{-1} C_p{}^T u^*$$

Colorization Based Coding Using GFT

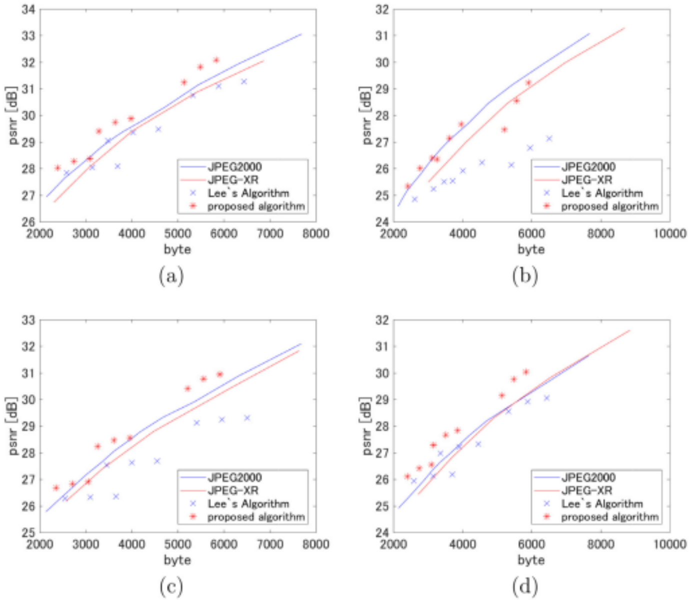
- Algorithm



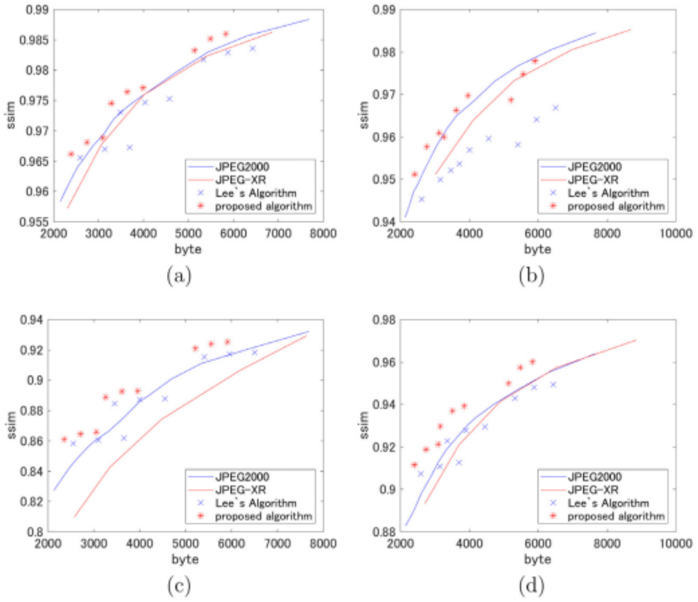
$u^* \in \mathbb{R}^{MN}$ $\hat{s} \in \mathbb{R}^{c}$ $u \in \mathbb{R}^{MN}$

Colorization Based Coding Using GFT

- PSNR



- SSIM



Graph Convolutional Network

- Graph neural network (GNN)=> deep learning on graph-structured data
  - Spatial-based GNN
  - Spectral-based GNN->GCN

- Convolutions on 2D signals (images)=>CNN

- How to define convolutions on graph signals?

Convolutions on graphs

- Convolution theorem: $f * g = F^{-1}\{F\{f\} \times F\{g\}\}$
- Spectral domain multiplication ⇔vertex domain convolution

$$
\begin{aligned}
y &= \mathrm{U} g_\theta(\Lambda) \mathrm{U}^\mathrm{T} x \\
&= g_\theta(\mathrm{U} \Lambda \mathrm{U}^\mathrm{T}) x \\
&= g_\theta(L) x
\end{aligned}
$$

Convolutions on graphs

$$g_\theta(\Lambda) = \sum_{k=0}^{K} \theta_k \Lambda^k \longrightarrow g_{\theta'}(\Lambda) = \sum_{k=0}^{K} \theta'_k T_k(\Lambda) \qquad (\Lambda = \frac{2\Lambda}{\Lambda_{\max}} - I,\ \Lambda \in [-1, 1])$$

$$y = g_{\theta'}(L)x = \sum_{k=0}^{K} \theta'_k T_k(L)\, x$$

T_k: Chebyshev polynomials (avoid direct computation of L)

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x),\ T_0(x) = 1,\ T_1(x) = x$$

Graph Convolutional Network

$$y = g_{\theta'}(L)x = \sum_{k=0}^{K} \theta_k' T_k(L)\, x, \quad K = 1$$

$$= \theta_0' x + \theta_1' L x$$

$$= \theta_0' x + \theta_1' \left(\frac{2L}{\lambda_{\max}} - I\right) x$$

$$= \theta_0' x + \theta_1' (L - I) x \qquad (\lambda_{\max} \approx 2)$$

$$= \theta_0' x + \theta_1' (D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) x \quad (L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}})$$

$$= \theta (I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) x \qquad (\theta = \theta_0' = -\theta_1')$$

$$I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \rightarrow D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

$$(A = A + I_N,\ D_{ii} = \sum_j A_{ij})$$

renormalization trick: avoid gradient vanishing/exploding

Graph Convolutional Network

1 GCN layer: $H^{(l+1)} = \sigma(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H^{(l)} W)$

nonlinear activation

define local neighborhood

input feature map

filter weight

=>correspond to 1 convolutional layer in CNN

Results

- Model: 2-layer GCN $Z = f(X, A) = \text{softmax}(A\ \text{ReLU}(AXW^{(0)})W^{(1)})$

Method	Citeseer	Cora	Pubmed	NELL
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

Outline

- Introduction
  - Graph Signal
  - Graph Shift
  - Graph Filter
  - Graph Fourier Transform (GFT)
  - Properties and Example
- Application
  - Image Coding using GFT
  - Graph Convolutional Network (GCN)
- Conclusion
- Reference

Conclusion

- GFT is a graph signal analysis approach. Like DFT in DSP, it can reflect frequencies by observing signals in the spectral domain.
- Colorization based coding is an idea of image coding, which utilizes colorization technique. Colorization based coding using GFT constructs a graph of representative pixels of an image and applies GFT to use values of spectrum for encoding.
- GCN is a type of graph neural network (GNN), it is based on spectral graph theory and models convolutions in the vertex domain as multiplications in the spectral domain.

Reference

[1] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs: Graph Fourier transform,” in Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP), 2013.
[2] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proc. ICLR, 2017, pp. 1–14.
[3] http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML2020/GNN.pdf
[4] L. Stankovi´c, M. Dakovi´c, and E. Sejdi´c, “Introduction to graph signal processing,” in *Vertex-Frequency Analysis of Graph Signals*, pp. 3–108, Springer, 2019.
[5]] K. Uruma, K. Konishi, T. Takahashi, T. Furukawa, Colorization-based image coding using graph Fourier transform, Signal Process., Image Commun. 74 (2019) 266–279.
[6] A. Levin, D. Lischinski, and Y. Weiss, “Colorization Using Optimization,” Proc. ACM Siggraph, 2004.
[7]] T. Ueno, T. Yoshida, and M. Ikehara, “Color image coding based on the colorization,” in Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific, Dec 2012, pp. 1–4.