Encoding Robust Representation for Graph Generation

Dongmian Zou, Gilad Lerman
University of Minnesota

Presenter: Arshdeep Sekhon
https://qdata.github.io/deep2Read

Outline

Graph Generation

- Most Graph Generation Methods use Graph Nets with VAE or GAN
- Require Training of Encoder-Decoder or Generator-Discriminator

Graph Generation

- Most Graph Generation Methods use Graph Nets with VAE or GAN
- Require Training of Encoder-Decoder or Generator-Discriminator
- Proposed Solution:
  SCAT: Deterministic Encoder using Scattering Transform so that only need to train decoder

Proposed Method

- $G = (V, E)$, $|V| = N$ vertices, $\mathbf{X} \in R^{N \times K}$
- Scattering Transform: $\mathbf{X} \Rightarrow \bar{\mathbf{X}}$
- Gaussianization : $\bar{\mathbf{X}} \Rightarrow \mathbf{z}$
- Decoder: $\mathbf{z} \Rightarrow \tilde{G}$

Graph Scattering Transform: $\mathbf{X} \Rightarrow \bar{\mathbf{X}}$

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{1}$$

The spectral decomposition :

$$\mathbf{L} = \Sigma_{l=0}^{N-1} \lambda_l \mathbf{u}_l \mathbf{u}_l^* \tag{2}$$

Fourier transform of a graph is defined as:

$$\mathcal{F}\boldsymbol{f} = \hat{\boldsymbol{f}} := (\boldsymbol{u}_l^* \boldsymbol{f})_{l=0}^{l=N-1} \tag{3}$$

$$\mathcal{F}^{-1}\hat{\boldsymbol{f}} := \sum_{l=0}^{N-1} \hat{\boldsymbol{f}}(l)\boldsymbol{u}_l \ . \tag{4}$$

$$\boldsymbol{f}_1 * \boldsymbol{f}_2 = \mathcal{F}^{-1}\left(\hat{\boldsymbol{f}}_1 \odot \hat{\boldsymbol{f}}_2\right) = \sum_{l=0}^{N-1} \boldsymbol{u}_l \hat{\boldsymbol{f}}_1(l)\hat{\boldsymbol{f}}_2(l) = \sum_{l=0}^{N-1} \boldsymbol{u}_l \boldsymbol{u}_l^* \boldsymbol{f}_1 \hat{\boldsymbol{f}}_2(l)$$

$=\sum_{l=0}^{N-1} \boldsymbol{u}_l \boldsymbol{u}_l^* \boldsymbol{f}_1 \boldsymbol{u}_l^* \boldsymbol{f}_2$.(5)

Wavelets

- signal transforms used in signal processing
- localize signal in both frequency and space
- constructed by translating and scaling a single “mother” wavelet
- difficult to do it directly on graphs: defining translation and scaling on graphs is hard
- Solution: wavelet transforms in spectral domain from Laplacian

Graph Wavelet Transform[1]

a scaling function ϕ and a wavelet functions for translation ψ with corresponding Fourier transforms $\hat{\phi}$ and $\hat{\psi}$. Dyadic Wavelets:

$$\hat{\psi}_j(\omega) = \hat{\psi}(2^{-j}\omega),\ j \in \mathbb{Z}\ . \tag{6}$$

$$\left|\hat{\phi}_{-J}\right|^2 + \sum_{j>-J} \left|\hat{\psi}_j\right|^2 = 1\ . \tag{7}$$

For $j > -J$, denote by $\hat{\boldsymbol{\psi}}_j$ the vector in $\mathbb{C}^N$ with the following entries:
$\hat{\boldsymbol{\psi}}_j(l) = \hat{\psi}_j(\lambda_l) = \hat{\psi}(2^{-j}\lambda_l),\ l = 0, \cdots, N-1.$ Similarly,
$\hat{\boldsymbol{\phi}}_{-J}(l) = \hat{\phi}_{-J}(\lambda_l) = \hat{\phi}(2^{-J}\lambda_l).$

[1] Graph Convolutional Neural Networks via Scattering
https://arxiv.org/pdf/1804.00099.pdf

Graph Wavelet Transform[1]

a scaling function ϕ and a wavelet functions for translation ψ with corresponding Fourier transforms $\hat{\phi}$ and $\hat{\psi}$. Dyadic Wavelets:

$$\hat{\psi}_j(\omega) = \hat{\psi}(2^{-j}\omega),\ j \in \mathbb{Z}\ . \tag{6}$$

$$\left|\hat{\phi}_{-J}\right|^2 + \sum_{j>-J} \left|\hat{\psi}_j\right|^2 = 1\ . \tag{7}$$

For $j > -J$, denote by $\hat{\boldsymbol{\psi}}_j$ the vector in $\mathbb{C}^N$ with the following entries: $\hat{\boldsymbol{\psi}}_j(l) = \hat{\psi}_j(\lambda_l) = \hat{\psi}(2^{-j}\lambda_l),\ l = 0, \cdots, N-1$. Similarly, $\hat{\boldsymbol{\phi}}_{-J}(l) = \hat{\phi}_{-J}(\lambda_l) = \hat{\phi}(2^{-J}\lambda_l)$. Replacing:

$$\boldsymbol{f} * \psi_j = \sum_{l=0}^{N-1} \boldsymbol{u}_l \boldsymbol{u}_l^* \boldsymbol{f} \hat{\psi}(2^{-j}\lambda_l) \text{ for } j > -J \text{ and } \boldsymbol{f} * \boldsymbol{\phi}_{-J} = \sum_{l=0}^{N-1} \boldsymbol{u}_l \boldsymbol{u}_l^* \boldsymbol{f} \hat{\phi}(2^J \lambda_l)\ . \tag{8}$$

[1] Graph Convolutional Neural Networks via Scattering
https://arxiv.org/pdf/1804.00099.pdf

Graph Wavelet Transform

For $\mathbf{f} \in R^N$

$$\mathbf{f} * \psi_j = \Sigma_{l=0}^{N-1} \mathbf{u}_l \mathbf{u}_l^* \mathbf{f} \hat{\psi}(2^{-j}\lambda_l) \text{ for } j > -J \tag{9}$$

$$\mathbf{f} * \phi_{-J} = \Sigma_{l=0}^{N-1} \mathbf{u}_l \mathbf{u}_l^* \mathbf{f} \hat{\phi}(2^{J}\lambda_l) \tag{10}$$

Graph Scattering Transform

The scattering transform for m no larger than the number of layers, a path $p = (j_1, j_2, \ldots, j_m)$, vector of m scales of the graph wavelets $0 \leq j_1, \ldots, j_m \leq J - 1$

$$\mathbf{S}[\mathcal{P}]\mathbf{f} = (\mathbf{S}[p]\mathbf{f})_{p \in \mathcal{P}} \tag{11}$$

$$\mathbf{S}[p]\mathbf{f} = (\mathbf{U}[p]\mathbf{f} * \phi_{-J}) \tag{12}$$

$$\mathbf{U}[p]\mathbf{f} = |||\mathbf{f} * \psi_{j_1}| * \psi_{j_2}| * \cdots * \psi_{j_m}| \tag{13}$$

For the K dimensional signal $X = [\mathbf{X}_1 | \ldots | \mathbf{X}_K] \in R^{N \times K}$

Graph Scattering Transform

If the set $\mathcal{P}$ has L elements, $M = LK$

$$\boldsymbol{S}[\mathcal{P}]\boldsymbol{X} = \bar{\mathbf{X}} \in R^{N \times M} \tag{14}$$

Why Graph Scattering Transform?

- invariant to permutations
- invariant to graph manipulations
- fixed transformation, no need to train the parameters

Gaussianization

We want to generate points using a Gaussian distribution so that the generator can generate novel points.

Whitening/Spherization

For T graphs:

$$\tilde{\boldsymbol{\mathcal{X}}} = \{\bar{\mathbf{X}}^{(t)}\}_{t=1}^{T} \in R^{T \times NM} \tag{15}$$

$$\boldsymbol{\mu} = \frac{1}{T} \sum_{t=1}^{T} \bar{\mathbf{X}}^{(t)} \tag{16}$$

$$\Sigma = \frac{1}{T} \sum_{t=1}^{T} (\bar{\mathbf{X}}^{(t)} - \boldsymbol{\mu})(\bar{\mathbf{X}}^{(t)} - \boldsymbol{\mu})* \tag{17}$$

Whitening Map **A**:

$$\boldsymbol{A}\tilde{\boldsymbol{\mathcal{X}}} = \boldsymbol{\Sigma}^{-1/2}(\bar{\mathbf{X}}^{(t)} - \boldsymbol{\mu}) \tag{18}$$

Makes samples uncorrelated with distribution close to normal with Identity covariance

Whitening/Spherization

For T graphs:

$$\tilde{\mathcal{X}} = \{\bar{\mathbf{X}}^{(t)}\}_{t=1}^{T} \in R^{T \times NM} \tag{15}$$

$$\mu = \frac{1}{T}\sum_{t=1}^{T}\bar{\mathbf{X}}^{(t)} \tag{16}$$

$$\Sigma = \frac{1}{T}\sum_{t=1}^{T}(\bar{\mathbf{X}}^{(t)} - \mu)(\bar{\mathbf{X}}^{(t)} - \mu)* \tag{17}$$

Whitening Map **A**:

$$\boldsymbol{A}\tilde{\mathcal{X}} = \boldsymbol{\Sigma}^{-1/2}(\bar{\mathbf{X}}^{(t)} - \mu) \tag{18}$$

Makes samples uncorrelated with distribution close to normal with Identity covariance

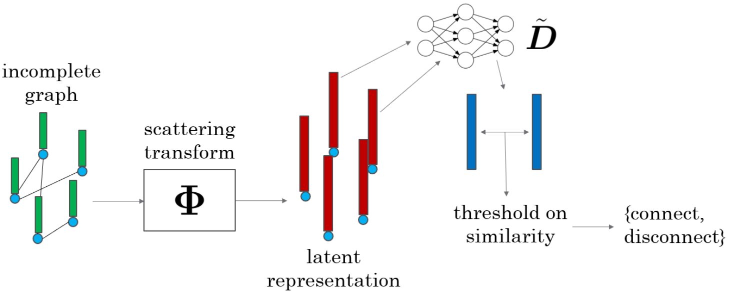
Spherization: Normalize samples to lie on unit sphere in Euclidean Norm

Output of Encoder and Gaussianization

$$z = \phi[\mathcal{P}](\boldsymbol{X}) \in R^{N \times M} \tag{19}$$

Decoder Task 1: Link Prediction

- One partially available graph
- encode the partially available graph $\mathbf{X} \in R^{N\times K}$ and $W_{train} \in R^{N\times N}$ into a latent vector $\mathbf{z} \in R^{N\times M}$
- No gaussianization
- $\boldsymbol{D(z)}=\sigma(\tilde{\mathbf{D}}(\mathbf{z})\tilde{\mathbf{D}}(\mathbf{z})^T)$
- Minimze the cross entropy loss
  $L(\mathbf{D}) = \sum_{i,j:\boldsymbol{W}(i,j)\neq 0}[-log\mathbf{D}(\phi(\boldsymbol{X},\boldsymbol{W}))(i,j)]$



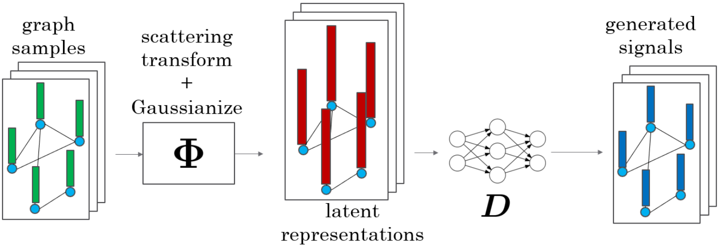
Decoder Task 2: Signal generation on graphs

- generating signals on a fixed graph

$$z = \phi[\mathcal{P}](\boldsymbol{X}) \in R^{N\times M} \tag{20}$$

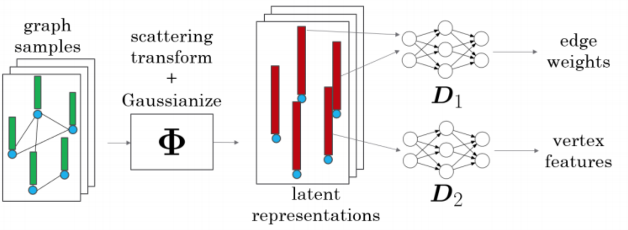
- Decode $\mathbf{D}(\mathbf{z})$ MLP $\in R^{N\times K}$
- Minimize Reconstruction Loss

$$L(D) = T^{-1}\sum_{t=1}^{T} ||\boldsymbol{X}^t - \boldsymbol{D}(\phi(\boldsymbol{X}^t))|| \tag{21}$$



Task 3: Generating graph and signals on the graphs

$$L(\boldsymbol{D}_1, \boldsymbol{D}_2) = T^{-1} \sum_{t=1}^{T} \Big[||\boldsymbol{W}^{(t)} - \boldsymbol{D}_1(\phi(\boldsymbol{X}^{(t)}))|| + ||\boldsymbol{X}^{(t)} - \boldsymbol{D}_2(\phi(\boldsymbol{X}^{(t)}))|| \Big] \tag{22}$$



Experiments

- simple Shannon wavelet
- the limiting scale $J = 3$
- For link prediction, use 2-layer scattering
- For signal and graph generation, use 3-layer scattering

Link Prediction

- links for the three citation datasets: Cora, Citeseer and Pubmed
- assume an undirected and unweighted graph

Dataset	Cora		Citeseer		Pubmed	
	AUC (%)	AP (%)	AUC (%)	AP (%)	AUC (%)	AP (%)
SCAT-S	94.48 ± 0.15	94.63 ± 0.17	97.27 ± 0.12	97.57 ± 0.12	97.52 ± 0.03	97.19 ± 0.04
SCAT-D	92.08 ± 0.09	93.05 ± 0.11	92.54 ± 0.14	94.16 ± 0.12	92.73 ± 0.17	93.56 ± 0.09
GAE	91.34 ± 0.52	92.62 ± 0.38	92.37 ± 0.67	93.72 ± 0.58	96.35 ± 0.18	96.53 ± 0.16
VGAE	91.14 ± 0.40	92.16 ± 0.29	92.70 ± 0.76	93.93 ± 0.57	95.68 ± 0.35	95.92 ± 0.32

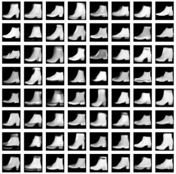
Dataset	SCAT-S	SCAT-D	GAE	VGAE
Cora	8.1ms	8.1ms	209.1ms	206.4ms
Citeseer	8.1ms	8.1ms	298.6ms	302.3ms
Pubmed	64.9ms	64.9ms	7832.6ms	7889.2ms

Signal Generation

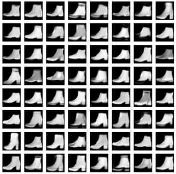
(a) Original data.

(b) SCAT-SW.

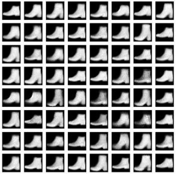
(c) SCAT-SN.



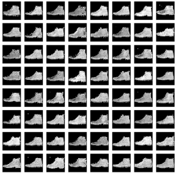
(d) SCAT-DW.



(e) SCAT-DN.



(f) VAE-GCN.



(g) GAN-GCN.

Signal and Graph generation

Algorithm	Valid	Unique	Novel
GraphVAE	(55.7)	(76.0)	(61.6)
GraphVAE (imp)	(56.2)	(42.0)	(75.8)
GraphVAE (no GM)	(81.0)	(24.1)	(61.0)
MolGAN (no RL)	90.4	31.1	97.8
MolGAN (RL Valid)	100.0	0.3	13.6
MolGAN (RL Unique)	99.2	37.1	64.5
MolGAN (RL Novel)	98.5	0.6	100.0
SCAT-SW	65.4	92.7	86.9
SCAT-DW	38.0	98.1	94.2
SCAT-SN	64.9	92.0	85.7
SCAT-DN	47.4	98.3	92.0

	SCAT-SW	SCAT-DW	SCAT-SN	SCAT-DN
Time (scattering)	137.12s	94.75s	132.01s	95.95s
Time (epoch)	3.68s	3.66s	3.71s	3.69s

Conclusions

- Deterministic Robust Encoder for Node Embeddings
- (+)Reduces training time
- (-)Requires fixed number of nodes
- (-)Other methods work better