From graph fourier to graph convolution– 1st spectral GCN and 2nd spectral GCN.

Mia Feng August 17, 2019

Fourier Transformation..

Traditional fourier transformation, known as $\mathcal{F}(w) = \mathscr{F}[f(t)] = \int f(t) e^{-iwt} = \hat{f}(t)$, is an integration of the mutiplication between signals $f(t)$ and basis function e^{-iwt}, while e^{-iwt} is the eigen function of Laplace operator.

Convolutional theorem..

The fourier transformation of convoluted functions equals covoluting functions transformed, aka $\mathscr{F}(f * h) = \hat{f} * \hat{h}$, where $*$ denotes convolution operation

Graph Fourier Transformation.

The Laplce matrix L of graph is defined as L = D − A, where D is degree matrix, and A is adjacency matrix.

Denote eigen vector of Laplace matrix L as U, then fourier transformation for graph is $F(\lambda_l) = \hat{f}(\lambda_l) = \sum_{i=1}^{N} f(i) u_l(i)$, aka

$$\begin{pmatrix} \hat{f}(\lambda_1) \\ \hat{f}(\lambda_2) \\ \vdots \\ \hat{f}(\lambda_N) \end{pmatrix} = \begin{bmatrix} u_1(1) & u_1(2) \cdots & u_1(N) \\ u_2(1) & u_2(2) \cdots & u_2(N) \\ \vdots & \vdots & \ddots \\ u_N(1) & u_N(2) \cdots & u_N(N) \end{bmatrix} \vdots \begin{pmatrix} f(\lambda_1) \\ f(\lambda_2) \\ \vdots \\ f(\lambda_N) \end{pmatrix} \tag{1}$$

, which can also be written as $\hat{f} = \mathrm{U}^{\mathrm{T}} f$.

The inverse fourier transformation is $f = \mathrm{U}^{\mathrm{T}} \hat{f}$.

Graph Convolution.

Denote h as a convolutional kernel, then the fourier tranformation of it is written as

$$\begin{pmatrix} \hat{h}(\lambda_1) & & \\ & \ddots & \\ & & \hat{h}(\lambda_N) \end{pmatrix}$$

, where $\hat{h}(\lambda_l) = \sum_{i=1}^{N} h(i)\, u_l(i)$.

The fourier transformation after convoluting f and h is

$$\begin{pmatrix} \hat{h}(\lambda_1) & & \\ & \ddots & \\ & & \hat{h}(\lambda_N) \end{pmatrix} \mathrm{U}^{\mathrm{T}} f$$

. But this convolutional results are measured in spectral space.

The convolutional results on a graph in spectral space is then recovered by reverse fourier transformation, which means multiplying U on the left. Therefore, it can be denoted as

$$(f * h) = \mathrm{U} \begin{pmatrix} \hat{h}(\lambda_1) & & \\ & \ddots & \\ & & \hat{h}(\lambda_N) \end{pmatrix} \mathrm{U}^{\mathrm{T}} f = \mathrm{U} \left(\mathrm{U}^{\mathrm{T}} h \bigodot \mathrm{U}^{\mathrm{T}} f \right) \tag{2}$$

, where $\bigodot$ denotes the multiplacation of matrix.

The point of convolutioning graph in spectral space is designing convolution kernel h.

1st GCN.

Kernel h in spectral space is calculated by BP directly, aka

$$y_o utput = \sigma \left(\mathrm{U} \begin{pmatrix} \theta(1) & & \\ & \ddots & \\ & & \theta(n) \end{pmatrix} \mathrm{U}^{\mathrm{T}} x \right) \tag{3}$$

, where x is a $n \times D$ matrix, n is the number of node, and D is the dimension of node features.

However, when graph is huge, factorizing U can be difficult.

2nd GCN.

Kernel h in spectral space is defined as $\hat{h}(\lambda_i) = \sum_{j=0}^{K} \alpha_j \lambda_i^j$, which is a multinomial approximation.

$$y_{output} = \sigma\left(\mathrm{U}\begin{pmatrix} \sum_{j=0}^{K} \alpha_j \lambda_1^j & & \\ & \ddots & \\ & & \sum_{j=0}^{K} \alpha_j \lambda_n^j \end{pmatrix} \mathrm{U}^{\mathrm{T}} x\right) \quad (4)$$

. Considering

$$\begin{pmatrix} \sum_{j=0}^{K} \alpha_j \lambda_1^j & & \\ & \ddots & \\ & & \sum_{j=0}^{K} \alpha_j \lambda_n^j \end{pmatrix} = \sum_{j=0}^{K} \alpha_j \Lambda^j \quad (5)$$

, then

$$\begin{aligned} y_{output} &= \sigma\left(\mathrm{U}\sum_{j=0}^{K} \alpha_j \Lambda^j \mathrm{U}^{\mathrm{T}} x\right) \\ &= \sigma\left(\sum_{j=0}^{K} \alpha_j \mathrm{U}\Lambda^j \mathrm{U}^{\mathrm{T}} x\right) \\ &= \sigma\left(\sum_{j=0}^{K} \alpha_j \mathrm{L} x\right) \end{aligned}$$

, where α_j is fine tuned by BP.

Parameter K perfoms as a paramter in spatial localization by narrowing the convolution in the K neighbouring nodes surrounding a center node.

The complexity of calculation is decreased, one is related to the size of kernel, the other is because no need of factorizing Laplace matrix.

Summary

The design of kernel is the key point of ocnvolution.