Semi-supervised classification with graph convolutional networks(GCN)

By Clement NSHIMIYIMANA

Overview

- Introduction
- Spectral Graph convolutions
- GCN
- Implementation
- Conclusion

Introduction

Idea

- Introducing CNN on graph structured data.
- Encode local graph structure and features of the nodes.
- Use scalable approach on semi supervised classification of nodes.
- Adopt first order approximation of spectral convolutions.

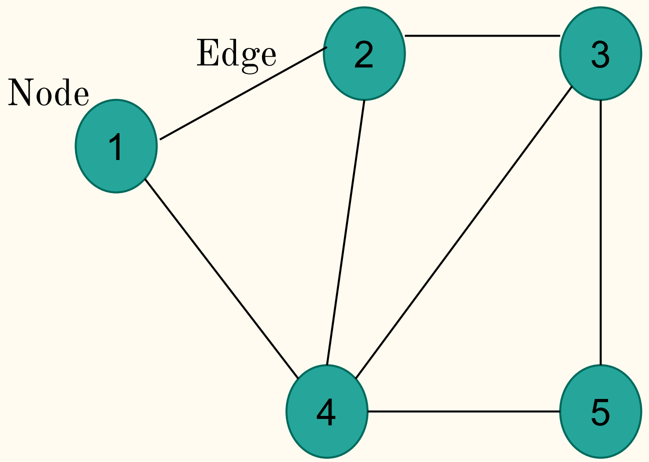
Graph

- G: (V, Є) with N nodes $v_i \in V$

  and $(v_i, v_j) \in$ Є .

- Adjacency matrix is matrix $A \in R^{NxN}$ representing the graph G.

- Degree matrix D is a diagonal matrix containing the information about degree of each node.

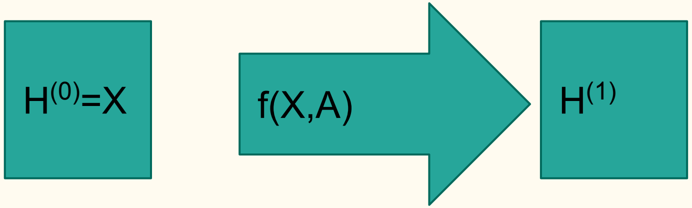


Convolutions on Graph

- Consider neural network function f(X, A)
- The layer wise propagation rule is

$$H^{(l+1)} = \sigma \left(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(l)} W^{(l)} \right)$$

- $\tilde{A} = A + I_N$, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$
- W: Weight matrix and σ: activation function like ReLu.



Spectral convolutions

- Spectral convolution on graph is the product of signal $x \in R^N$ and filter g_Θ:

$$g_\Theta * x = U g_\Theta U^T x \qquad (1)$$

- U: matrix of eigenvectors of of the normalized graph Laplacian L.
- $L = I_N - D^{-1/2} A D^{-1/2}$
- $U^T x$: graph fourier transform of x
- g_Θ : diagonal matrix parametrized by $\Theta \in R^N$

- As computing eq(1) is expensive, then we can approximate it by a truncated expansion

$$g_\theta(\Lambda) \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{\Lambda}) \quad (2)$$

- $\tilde{\Lambda} = (2/\lambda_{max})\Lambda - I_N$, with λ_{max} denotes the largest eigenvalue of L.

- $T_k(\tilde{\Lambda}) = 2\tilde{\Lambda}T_{k-1}(\tilde{\Lambda}) - T_{k-2}(\tilde{\Lambda})$ with $T_0(\tilde{\Lambda}) = 1$ and $T_1(\tilde{\Lambda}) = \tilde{\Lambda}$: Chebyshev polynomials.

- The equation (1) becomes $g_\theta * x = \sum_{k=0}^{K} \theta_k T_k(\tilde{L})\ x \qquad (3)$
- It is denoted K^{th}-order polynomial in the Laplacian, with $\tilde{L} = (2/\lambda_{max})L - I_N$.

GCN

- GCN is linear scale of equation (3)
- k=1, λ_{max}=2, $g_\theta * x \approx \theta_0 x + \theta_1 (L - I_N) x$

$$= \theta_0 x - \theta_1 D^{-1/2} A D^{-1/2} x$$

- In practice, $g_\theta * x \approx \theta(I_N + D^{-1/2} A D^{-1/2})x$, $\theta = \theta_0 = \theta_1$
- In NN to avoid exploding/vanishing gradient we consider $\tilde{A} = A + I_N$

$$\boxed{Z = g_\theta * x = (\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}) x \theta}$$

$\theta \in R^{C \times F}$: matrix of filter parameters

$Z \in R^{N \times F}$: convolved signal

C : input channels, F: filters.

Implementation

- We consider a two-layer GCN for semi-supervised node classification on a graph.

- Semi-supervised document classification in citation networks.

- Datasets: Citeseer, Cora, Pubmed.

Results

	Citeseer	Cora	Pubmed
GCN(paper)	67.9	80.1	78.9
GCN(me)	68.1	80.8	78.7

Conclusion

- GCN model uses an efficient layer-wise propagation rule that is based on a first-order approximation of spectral convolutions on graphs.

- GCN model is capable of encoding both graph structure and node features in a way useful for semi-supervised classification.

- GCN outperform other method like MLP.

- 
Thank You