2. Semi-Supervised Classification with Graph Convolutional Networks

이은경

Graduate School, Dept. of Urban Big Data Convergence

2021

Outline

1. Introduction I

- problem : classifying nodes in a graph. ¿ node classification
- 라벨이 어느정도만 있고, 나머지는 없는 데이터의 경우에 GCN 을 어떻게 적용시킬지에 대한 모델 ¿ semi-supervised learning
- graph laplacian regurlarization term in the loss function
- $\mathcal{L} = \mathcal{L}_0 + \lambda \mathcal{L}_{reg}, with \mathcal{L}_{reg} = \sum_{i,j} A_{ij} \|f(X_i) - f(X_j)\|^2 = f(X)^T \triangle f(X)$
- $\mathcal{L}_0$ 은 라벨이 있는 부분에 사용하는 supervised loss
- f(.) 은 신경망에서 미분가능한 함수처럼 사용 가능한 함수.
- λ 는 weight factor
- X는 노드의 feature vectors matrix
- $\triangle$ 은 D-A로 방향이 없는 그래프에서 라플라시안 비표준화를 나타냄.
- N은 노드 $v_i \in \mathcal{V}$
- 엣지는 $(v_i, v_j) \in \mathcal{E}$

1. Introduction II

- adjacency matrix 는 $A \in R^N \times N$
- degree matrix $D_{ii} = \sum_j A_{ij}$
- 1) 그래프에서 직접 작동하는 신경망 모델에 대해 간단하고 성능좋은 계층별 전파 규칙 수행 : 스펙트럼 그래프 컨볼루션의 1차 근사에서 영감 (Hammond et al., 2011).
- 2) 그래프 기반 신경망 모델이 어떻게 사용될 수 있는지, 빠르고 확장 가능한 semi-supervised 방식의 그래프 node 분류.
- ex. Citation network

1. Introduction III

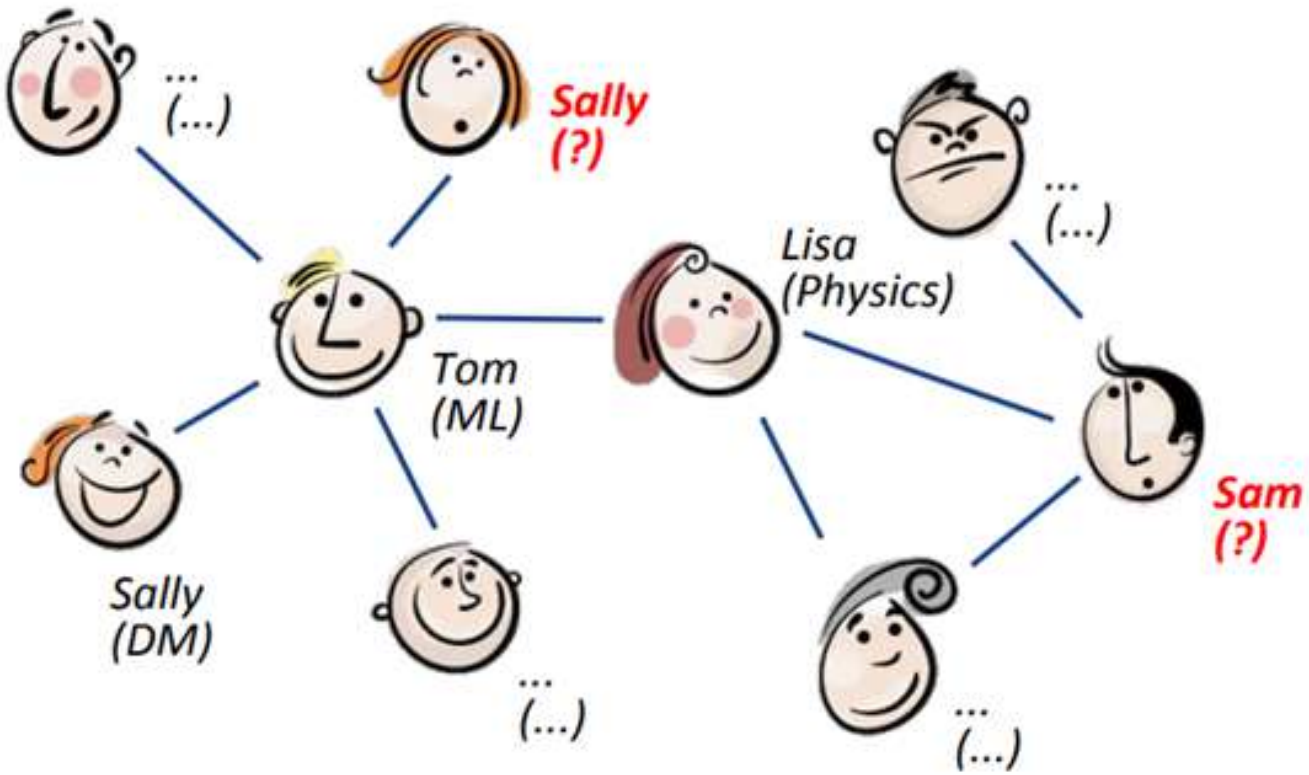


Figure: Spatial to Spectral

2. Fast Approximate Convolutions On Graphs I

- a multi-layer Graph Convolutional Network (GCN) with the following layer-wise propagation rule: $H^{(l+1)} = \sigma(\tilde{D}^{\frac{-1}{2}}\tilde{A}\tilde{D}^{\frac{-1}{2}}H^{(l)}W^{(l)})$
- $\tilde{A} = A + I_N$ adjacency matrix of the undirected graph G with added self-connections.
- I_N : the identity matrix
- $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, $W^{(l)}$: layer-specific trainable weight matrix
- $\sigma(\cdot)$: activation function
- $ReLU(\cdot) = max(0, \cdot)$
- $H^{(l)} \in \mathbb{R}^{N \times D}$: matrix of activations in the l-th layer
- $H^{(0)} = \mathrm{X}$

2.1. SPECTRAL GRAPH CONVOLUTIONS I

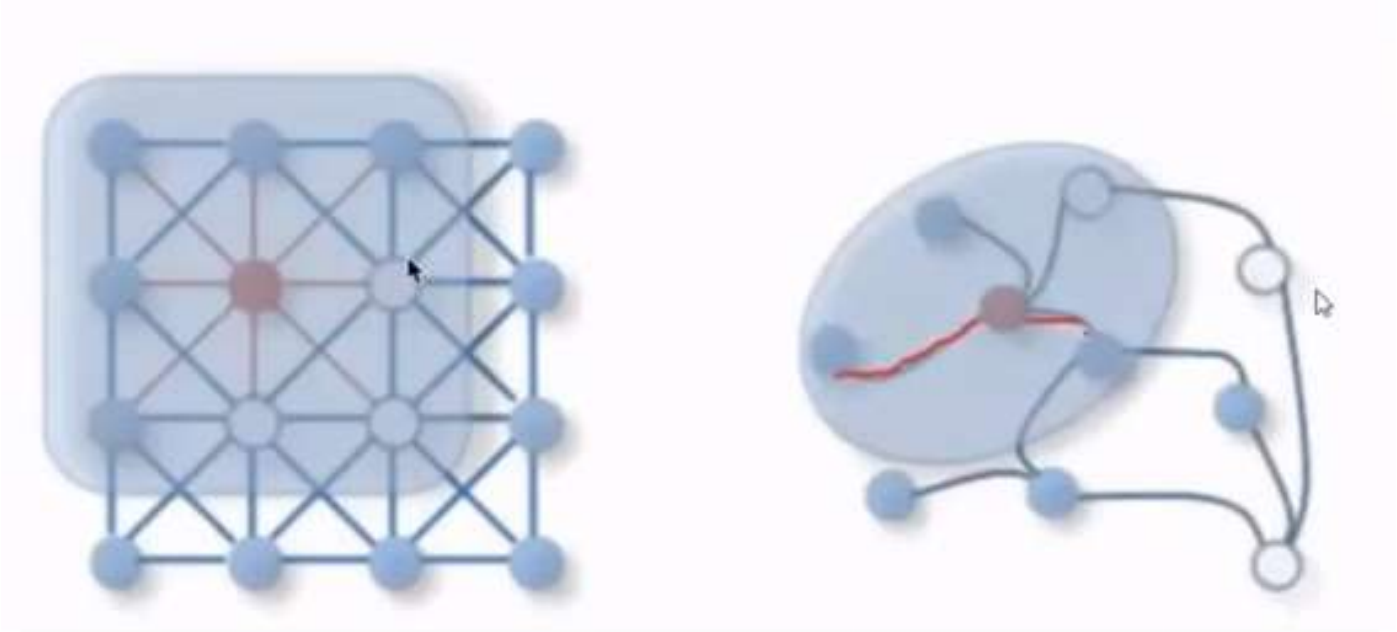


Figure: Spatial to Spectral

2.1. SPECTRAL GRAPH CONVOLUTIONS II

- spectral 정의 : spatial을 그래프의 pre-transpormation 로 형변환을 해서 그래프의 신호나 변화를 주었을 때 얼마나 많이 변동이 일어나는지 측정하는 방식
- spectral convolutions on graphs : def) the multiplication of a signal $x \in \mathbb{R}^N$ (모든 노드의 스칼라) with a filter
  - 라플라시안을 적용하면, 각각의 vertex 가 가지고 있는 에너지의 양까지 볼 수 있음. 라플라시안 – 연결성.변화 표현 가능

2.1. SPECTRAL GRAPH CONVOLUTIONS III

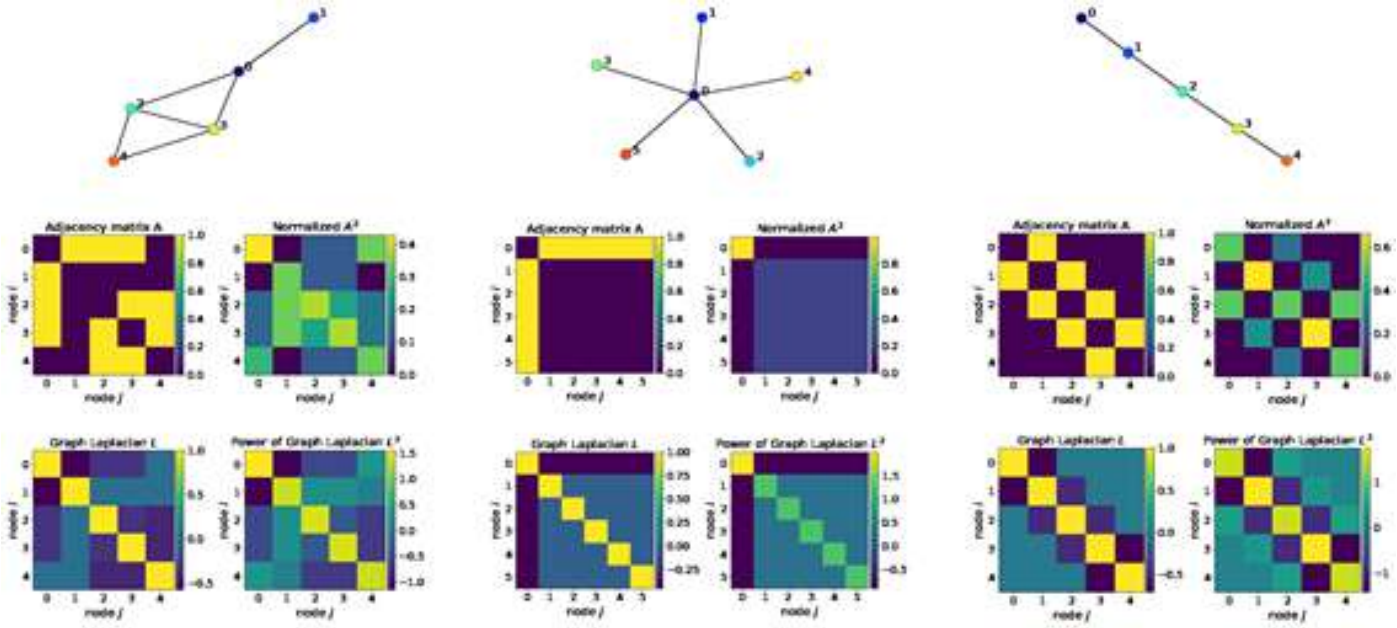


Figure: Spatial to Spectral

2.1. SPECTRAL GRAPH CONVOLUTIONS IV

- 푸리에 domain 에서 $\theta \in \mathbb{R}^N$, 필터는 $g_\theta = diag(\theta)$
- i.e. $g_\theta \times x = U_{g_\theta} U^T x \cdots (1)$

- U : 그래프 라플라시안 정규화된 고유벡터 행렬.
  $L = I_N - D^{\frac{-1}{2}} A D^{\frac{-1}{2}} = U \Lambda U^T$
- Λ : 고유값들의 대각행렬, $U^T x$: x의 그래프에 대한 푸리에 변환 값, g_θ : L의 고유값에 대한 함수
- (1) 은 계산량이 많음. 고유벡터 행렬 U를 계산하는데 복잡도는 $\mathcal{O}(N^2)$. 그래프의 규모가 커질수록 이 계산량은 폭발
- 이 문제를 우회하기 위해 체비셰프 다항식 측면에서 확장.
  $g_{\theta'}(\Lambda) \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda})$. 여기서 $\tilde{\Lambda} = \frac{2}{\lambda_{max}} \Lambda - I_N$. λ_{max} : L 에 대한 최대 고유값. $\theta' \in \mathbb{R}^K$ 는 체비셰프 계수에 대한 벡터. 체비셰프 다항식은 재귀적으로 정의. $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, $T_0(x) = 1, T_1(x) = x$

- 이러한 체비셰프 다항식을 spectral convolutions on graphs 에 적용하면
  $g_{\theta'} \times x \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{L}) x \cdots (2)$

2.1. SPECTRAL GRAPH CONVOLUTIONS V

- $\tilde{L} = \frac{2}{max} L - I_N$. 이는 $(U \Lambda U^T)^k = U \Lambda^k U^T$
- 이 표현식은 라플라시안에서 K차 다항식이기 때문에 중앙 노드에서 최대 K 단계 떨어진 노드까지만 의존하도록 K-localized 시킴.((K th-order neighborhood)
- 이로 인해 복잡도는 $\mathcal{O}([\mathcal{E}])$, 즉 모서리 수가 선형. K-localized convolution을 사용하여 그래프에 convolutional 신경망을 정의함.

2.2. LAYER-WISE LINEAR MODEL I

- 그래프 컨볼루션을 기반으로하는 신경망 모델은 여러 형태의 컨벌루션 레이어가 쌓인 구조로 구성. 각 레이어는 point-wise 비선형성을 띔.
- 예) 레이어별 컨볼루션 연산을 $K = 1$로 제한했다고 가정하면 선형 w.r.t. L 따라서 그래프 라플라시안 스펙트럼의 선형 함수.
- 이런 식으로, 여러 층을 쌓아서 풍부한 종류의 컨볼루셔널 필터 함수를 사용 가능. 그러나 Chebyshev 다항식에 의해 주어진 명시적 매개 변수화에 제한되지 않음. 우리는 이러한 모델이 과적합 문제를 완화할 수 있다고 기대할 수 있음.
- 매우 넓은 노드 그래프 분포를 가진 그래프에 대한 K-localized 네트워크 구조 일때도 적용가능 e.g. social networks, citation networks, knowledge graphs and many other real-world graph datasets
- 고정된 계산 예산의 경우이 layer-wise 선형 변환을 사용하면 여러 도메인에서 모델링 능력을 향상시키는 것으로 알려짐.

2.2. LAYER-WISE LINEAR MODEL II

- 이 GCN에서의 선형변환 형태는 이러한($max \approx 2$) 근사치에서 예상 할 수 있듯이 네트워크 매개 변수는 훈련 중에 이러한 규모의 변화에 적응함.
$g_{\theta'} \times x \approx \theta'_0 x + \theta'_1 (L - I_N)x = \theta'_0 x - \theta'_1 D^{\frac{1}{2}} A D^{\frac{-1}{2}} x \cdots (3)$

- 두 개의 자유 매개변수 θ'_0, θ'_1
- 필터 매개변수는 전체 그래프에서 공유. 이 형식의 필터를 연속적으로 적용하면 다음의 k 차 이웃 노드가 효과적으로 컨볼 루션됩니다.
- k : 신경망 모델에서 연속적인 필터링 연산 또는 컨벌루션 레이어의 수

2.2. LAYER-WISE LINEAR MODEL III

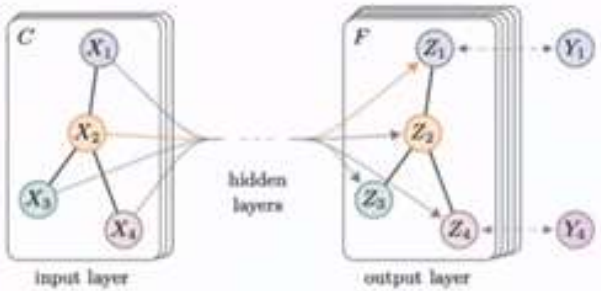
- 실제로 과적합 문제를 해결하기 위해 매개변수 수를 제한하는 것이 유용할 수 있음. 계층당 연산(예 : 행렬 곱셈)수를 최소화.
  $g_{\theta'} \times x \approx \theta(I_N + D^{\frac{-1}{2}} A D^{\frac{-1}{2}})x \cdots (4)$
  - 단일 파라메터 $\theta = \theta'_0 = -\theta'_1$. $I_N + D^{\frac{-1}{2}} A D^{\frac{-1}{2}}$: [0,2] 범위 내에서 고유값을 가짐. 따라서 이 연산자를 반복적으로 적용하면 수치가 불안정해질 수 있음. 심층 신경망 모델에서 사용되는 경우 그레디언트의 폭발/소멸 문제 발생. 이를 완화하기 위해 재정규화 트릭 사용 $I_N + D^{\frac{-1}{2}} A D^{\frac{-1}{2}} \rightarrow \tilde{D}^{\frac{-1}{2}} \tilde{A} \tilde{D}^{\frac{-1}{2}}$. with $\tilde{A} = A + I_N$ and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$
- 이 정의를 신호로 일반화 할 수 있음. $X \in \mathbb{R}^{N \times C}$ with C : input channels (C-dimensional Feature Vector for every node) and F : feature or filter $Z = \tilde{D}^{\frac{-1}{2}} \tilde{A} \tilde{D}^{\frac{-1}{2}} X \Theta$. where $\Theta \in \mathbb{R}^{C \times F}$ 은 필터 파라메터의 행렬이고, $Z \in \mathbb{R}^{N \times F}$ 은 복잡한 신호 행렬.
  - 이 필터 연산의 복잡도는 $\mathcal{O}([\mathcal{E}]FC)$, as $\tilde{A}X$ 밀도가 높은 행렬이 있는 희소 행렬의 곱으로 효율적으로 구현 가능하다.

3. SEMI-SUPERVISED NODE CLASSIFICATION I

- 효율적인 정보 전파를 위해 간단하면서도 유연한 모델 $f(X, A)$를 도입해 그래프 기반 준지도 학습에서 일반적으로 사용되는 특정 가정을 완화할 수 있음. 데이터 X와 인접 행렬 A모두에서 모델 $f(X, A)$를 조정.
- 그래프에서 semi-supervised로 노드 분류를 위한 2-layer GCN을 고려. 대칭 인접 행렬 A (이진 또는 가중)를 사용 $\hat{A} = \tilde{D}^{\frac{-1}{2}}\tilde{A}\tilde{D}^{\frac{-1}{2}}$ 를 전처리 단계에서 계산.
- forward model : $Z = f(X, A) = softmax(\hat{A}ReLU(\hat{A}XW^{(0)})W^{(1)})$

3. SEMI-SUPERVISED NODE CLASSIFICATION II

GCN Layers (Model)



(a) Graph Convolutional Network

(b) Hidden layer activations

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$$

$$Z = f(X, A) = \mathrm{softmax}\Big(\hat{A}\ \mathrm{ReLU}\Big(\hat{A}XW^{(0)}\Big)\,W^{(1)}\Big)\ .$$

$$L = -\sum_{l \in Y_L}\sum_{f=1}^{F} Y_{lf} \ln Z_{lf}$$

Y: 실제 라벨 정답, Z: 예측 값 에 대한 cross-entropy

Figure: GCN structure

3. SEMI-SUPERVISED NODE CLASSIFICATION III

- $W^{(0)} \in \mathbb{R}^{C \times H}$ 은 H feature map 의 hidden layer 에 대한 input-to-hidden weight 행렬
- $W^{(1)} \in \mathbb{R}^{H \times F}$ 은 hidden-to-output weight 행렬
- softmax 를 activate function 으로 사용. $softmax(x_i) = \frac{1}{Z} exp(x_i)$ with $Z = \sum_i exp(x_i)$ row-wise 적용.

- 모든 라벨링된 examples에서 cross-entropy error 계산을 위한 Loss 함수는 $\mathcal{L} = -\sum I \in \dagger_L \sum_{f=1}^{F} Y_{lf} InZ_{lf}$ where $\mathcal{Y}_L$은 라벨이 있는 노드들의 set.
- 신경망 가중치 $W^{(0)}$ 및 $W^{(0)}$은 경사 하강법을 사용하여 훈련. 모든 훈련 반복에 대해 전체 데이터 세트를 사용하여 배치 경사 하강법을 수행

4. Experiments I

Table 1: Dataset statistics, as reported in Yang et al. (2016).

Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001

Figure: Spatial to Spectral

4. Experiments II

- Citation networks : 각 문서 및 목록에 대한 희소 bag-of-words 특징 벡터 문서 간 인용 링크 수.
- NELL : 지식 그래프에서 추출된 데이터, 지식 그래프는 방향이 있고 레이블이 지정된 관계와 연결된 엔티티 집합
- Random graphs : 실험을 위해 다양한 크기의 랜덤 그래프 데이터 세트를 시뮬레이션한 데이터

5. Results I

Table 2: Summary of results in terms of classification accuracy (in percent).

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

Figure: Spatial to Spectral

5. Results II

Table 3: Comparison of propagation models.

Description		Propagation model	Citeseer	Cora	Pubmed
Chebyshev filter (Eq. 5)	$K = 3$	$\sum_{k=0}^{K} T_k(\tilde{L})X\Theta_k$	69.8	79.5	74.4
	$K = 2$		69.6	81.2	73.8
1st-order model (Eq. 6)		$X\Theta_0 + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}X\Theta_1$	68.3	80.0	77.5
Single parameter (Eq. 7)		$(I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})X\Theta$	69.3	79.2	77.4
Renormalization trick (Eq. 8)		$\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta$	**70.3**	**81.5**	**79.0**
1st-order term only		$D^{-\frac{1}{2}}AD^{-\frac{1}{2}}X\Theta$	68.7	80.5	77.8
Multi-layer perceptron		$X\Theta$	46.5	55.1	71.4

Figure: Spatial to Spectral