SIGN : Scalable Inception Graph Neural Networks

Intro.

- GNNs는 classical 한 CNNs을 graph structured data에 적용시키려는 연구방향으로 다양한 convolution-like operations가 제안되었다.
- GNNs에서 scaling은 graph deep learning을 산업 전반에서 적용시키려는데 있어서 주된 장벽 중 하나이다.
- Euclidean neural networks 들은 loss에 대한 학습을 개별 샘플들에 나누어 볼 수 있는 반면에, graph convolutional networks는 그래프에서 edges를 따라 nodes 사이사이로 정보가 전달되기 때문에 loss computation이 다른 nodes에 대하여 interdependent 하게 된다.
- 게다가 filter의 receptive field가 증가하면서 nodes의 수도 지수적으로 늘어날 수 있다.
- 이런 문제를 다루기 위해 지금까지는 *graph sampling* 접근 방법이 많이 제안되어 왔다. 이는 작은 수의 이웃 nodes를 선택함으로써 GNN 학습 cost를 완화하였다.
- 이 논문에서는 다른 접근 방법으로 graph에서의 scalable deep learning을 핸들링하려고 한다.
- 제안된 architecture는 SIGN (simple scalable Graph Neural Network)이다.
- SIGN은 다양한 type과 size의 graph convolutional filters를 합쳐 사용하는데, precomputaion이 가능하여, 빠른 학습과 추론 속도를 가질 수 있다.
- SIGN 단 하나의 graph convoluitonal layer 만으로 몇몇 large-scale graph learning datasets에서 SOTA를 보였다.
- 우리가 생각해봐야하는 것은 "언제 deep graph neural network architectures 가 유용한지 특히, 언제 scalability 가 중요한 요구조건인지?"인데
- 최근엔 deep Graph Neural Networs를 위한 연구가 많이 이루어졌지만, 'small-world' networks와 같은 일반적인 불규칙 그래프에서는 depth의 이점이 없다고한다.
- 저자는 계속 depth를 키워서 성능을 높이는 방향이 아니라 국소적으로 더욱 표현력이 뛰어난 operators 연구에 초점을 맞추어야한다고 주장한다.

2 Background and Related Work

2.3 Convolution-like operators on graphs

S-GCN

- non-linearity σ 를 element-wise로 적용시킨 L개의 GCN layers를 쌓는 것은 넓은 receptive fields를 갖도록 합니다.

  $$\mathbf{Y} = \xi(\tilde{A} \cdots \sigma(\tilde{\mathbf{A}}\mathbf{X}\mathbf{\Theta}^{(1)}) \cdots \Theta^{(L)})$$

- *simplified* GCN(S-GCN) 논문은 large filters를 가지는 Graph convolutions는 실질적으로 작은 filters의 multiple convolutional layers 와 같다는 주장을 하였다.
- 그리고 마지막 layer를 제외하고 모든 중간 layers에 non-linearity 를 제거해도 성능에 큰 문제가 없음을 보였다.

  $$\mathbf{Y} = \xi(\tilde{\mathbf{A}}\mathbf{X}\mathbf{\Theta}^{(1)} \cdots \Theta^{(L)}) = \xi(\tilde{\mathbf{A}}^{\mathbf{L}}\mathbf{A}\mathbf{\Theta})$$

3 Scalable Inception Graph Neural Networks

- 핵심 building bolck은 n x n matrices $\mathbf{A_1}, \ldots, \mathbf{A_r}$ 로 표현되는 linear diffusion operators set이다. 이 행렬들로node-wise features를 pre-computed한다.
- node-wise classification taks를 가정하면 아래와 같다.

$$\mathbf{Z} = \sigma([\mathbf{X\Theta_0}, \mathbf{A_1X\Theta_1}, \ldots, \mathbf{A_rX\Theta_r}])$$

$$\mathbf{Y} = \xi(\mathbf{Z\Omega})$$

  - $\mathbf{\Theta_0}, \ldots, \mathbf{\Theta_r}, \Omega$ 는 learnable matrices이다. 차원은 각각 d x d' 과 d'(r+1) x c 이다. 여기서 c는 class 개수이다.
  - σ, ξ 는 각각 non-linearities이다.
  - 아래 식은 calss에 대한 확률을 계산하는 것으로 softmax나 sigmoid를 경우에 맞게 사용하면 된다.
  - Operators r 의 개수에 따라 SIGN-r 로 표현한다.
- 여기서 주의해서 살펴볼 것은 $\mathbf{A_1X}, \ldots, \mathbf{A_rX}$ 가 learnable model parameters에 depend하지 않는 다는 것이고 미리 계산이 가능하다는 것이다.
- 예를 들어 large graphs에 대해서 Apache Spark와 같은 분산처리 인프라스트럭쳐로 빠르게 계산이 가능하다.

Inception-like module

- SIGN framework에서 특정 operator $\mathbf{B}$ 를 선택하고 $\mathbf{A_k} = \mathbf{B}^k$ 로 정의할 수 있다. 이때 k = 1 to r
- 이러한 setting은 CNNs에서 Inception module의 방식을 차용한것이다.
  파라미터 r에 따라 filters의 size가 달라지는데, 여기서 r = 0 은 1 x 1 convolutions와 같은 의미이고
  이는 노드 개별 적으로 어떠한 diffusion 없이 featrues를 linear transformation 하는 것이다.
- 이러한 유사성을 따라서 저자는 본 논문에서 제안한 모델의 이름을 SIGN으로 했다고 한다.

Choice of the operators

- diffusion operators를 선택하는 것은 task, graph structure 그리고 featrues에 달려있다.
- 본 논문 실험에서는 세 종류의 operators를 사용하였는데,
  1. simple (normalized) adjacency matrix
  2. Personalized PageRank-based adjacency matrix
  3. Triangle-based adjacency matrix
- operatros를 조합할때 각각 p, s, t 개 사용하여 SIGN(p,s,t)로 표현하였고, 여기서 r = p+s+t 이다.

SIGN and S-GCN

- SIGN과 S-GCN은 본질적으로 "shallow" 하다. 일반적인 방식인 graph convolutional laeyrs를 연속적으로 stacked 하는 것이 아니라 single linear filtering operation으로 통합하거나 (S-GCN) 또는 평행하게 적용하여 multi-scale node representations를 계산하여 operators에 따라 다양한 패턴을 잡아낸다.(SIGN)
- S-GCN은 SIGN-1의 특수한 형태로 고려될 수 있다.

5. Conclusion and Future Work

Depth vs. widht for Graph Neural Networks

- Deep Graph Neural Networks는 vanishing gradients와 featrue smoothing 때문에 학습이 어렵다.
- 물론 이런 문제를 극복하려는 연구와 시도들이 있지만 depth가 무조건 shallow baselines의 퍼포먼스보다 항상 더 좋은 성능을 보장하진 못 한다.

Limitations

- 이 모델은 **BX** 와 같은 lienar graph aggregation operations를 사용함으로써 precomputation이 가능하고 이는 큰 장점이 된다.
- graph attention 기반의 방법들은 데이터에 적응적인 operator $\mathbf{B}_\theta(\mathbf{X})$ (θ 는 learable parameters) 를 이용하게 되는데, 이는 precomputaion의 이점을 포기해야한다.
- 한 가지 제안한 방법으로 먼저 작은 subset으로 Attention을 계산하고 이 operator를 고정한 다음 학습과 추론 할때 precompute 하여 적용하는 방법이 있다.