Inductive Representation Learning on Large Graphs

By William Hamilton, Rex Ying, and Jure Leskovec – NIPS’17

Presented by Aida Sheshbolouki
CS-848 Winter’19
Graphs
Complex networks

- Technological Networks
- Social Networks
- Infrastructural Networks
- Biological Networks
- ...

Graphs
Analytic Tasks

- Node classification
- Link prediction
- Community detection
- Network similarity
- ...

Graphs
Machine Learning Life Cycle

- Raw Data
- Structured Data
- Learning Algorithm
- Model

- Feature Engineering
- Downstream Prediction Task
Machine Learning Life Cycle

Raw Data → Structured Data → Learning Algorithm → Model

Feature Engineering

Automatic Feature Learning

Downstream Prediction Task
Node/Subgraph $\rightarrow$ A point in low dimensional vector space
Encoder-Decoder Perspective

Encoder Function

\[ \text{ENC} : \mathcal{V} \rightarrow \mathbb{R}^d \]

Decoder Function

\[ \text{DEC} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+ \]

Similarity Function

\[ \text{DEC}(\text{ENC}(v_i), \text{ENC}(v_j)) = \text{DEC}(z_i, z_j) \approx s_g(v_i, v_j) \]

Loss Function

\[ \mathcal{L} = \sum_{(v_i, v_j) \in \mathcal{D}} \ell(\text{DEC}(z_i, z_j), s_g(v_i, v_j)) \]
# Shallow Embedding

\[ \text{ENC}(v_i) = Zv_i \]

## Limitations:

1. No shared Parameters \( \rightarrow O(|V|) \) parameters
2. Transductive
3. Not leveraging nodes’ features
Neighborhood Encoders

**Key idea:** Generate node embeddings based on local neighborhoods.

**Intuition:** Nodes aggregate information from their neighbours using neural networks.
1) Define a neighborhood aggregation function.

2) Define a loss function on the embeddings, $\mathcal{L}(z_u)$
Neighborhood Encoders

3) Train on a set of nodes, i.e., a batch of compute graphs
4) Generate embeddings for nodes as needed

Even for nodes we never trained on!!!!
Neighborhood Aggregation

Basic Neighborhood Aggregation

\[ h^k_v = \sigma \left( \sum_{u \in N(v)} \frac{h^{k-1}_u}{|N(v)|} + B_k h^{k-1}_v \right) \]

vs.

GCN Neighborhood Aggregation

\[ h^k_v = \sigma \left( \sum_{u \in N(v) \cup v} \frac{h^{k-1}_u}{\sqrt{|N(u)||N(v)|}} \right) \]

same matrix for self and neighbor embeddings

per-neighbor normalization
GraphSAGE

Any differentiable function that maps set of vectors to a single vector.

\[ h_v^k = \sigma \left( [A_k \cdot \text{AGG} \left( \{h_u^{k-1}, \forall u \in \mathcal{N}(v)\} \right), B_k h_v^{k-1}] \right) \]
GraphSAGE Differences

- Simple neighborhood aggregation:

\[
\mathbf{h}_v^k = \sigma \left( W_k \sum_{u \in N(v)} \frac{h_{u}^{k-1}}{|N(v)|} + B_k h_v^{k-1} \right)
\]

- GraphSAGE:

\[
\mathbf{h}_v^k = \sigma \left( [W_k \cdot \text{AGG} (\{h_{u}^{k-1}, \forall u \in N(v)\})], B_k h_v^{k-1} \right)
\]

concatenate self embedding and neighbor embedding

generalized aggregation

Neighborhood sampling
GraphSAGE Functions

- **Mean**: \( \text{AGG} = \sum_{u \in N(v)} \frac{h_{u}^{k-1}}{|N(v)|} \)
- **Pool**: \( \text{AGG} = \gamma \left( \{ Qh_{u}^{k-1}, \forall u \in N(v) \} \right) \)
- **LSTM**: \( \text{AGG} = \text{LSTM} \left( [h_{u}^{k-1}, \forall u \in \pi(N(v))] \right) \)

**Loss function**

\[
J_{G}(z_{u}) = - \log \left( \sigma(z_{u}^{T}z_{v}) \right) - Q \cdot \mathbb{E}_{v_{n} \sim P_{n}(v)} \log \left( \sigma(-z_{u}^{T}z_{v_{n}}) \right)
\]
GraphSAGE

An inductive encoder
Parameter sharing
Integrating the graph structure and node feature
References

- Representation Learning on Networks, snap.stanford.edu/proj/embeddings-www, WWW 2018

Discussion

- GraphSAGE is optimized for two or three layers, what if we want to go deeper? What are the challenges?

- How can we extend GraphSAGE to support multi-layer networks?

- GraphSAGE generates embedding for nodes, what about subgraph embedding?