Tencent AI Lab

Representation Learning on Graphs

Yu Rong

Machine Learning Group

Agenda

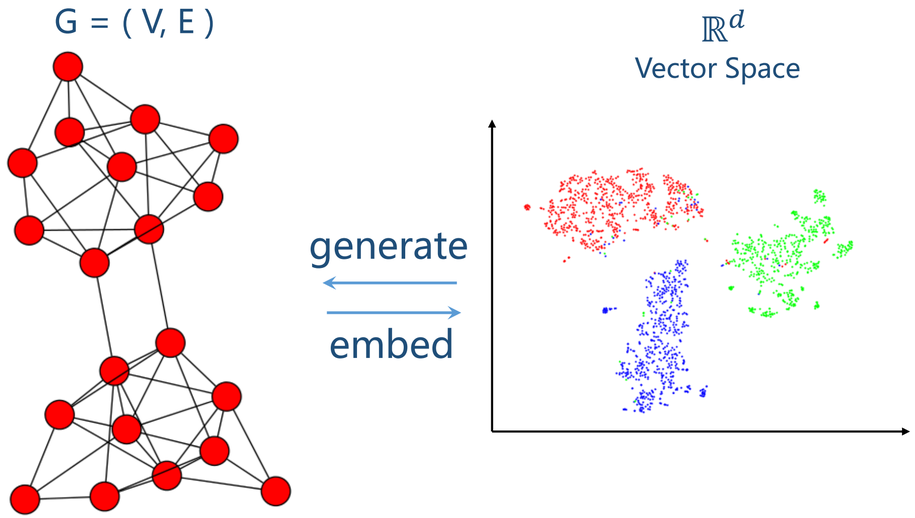
Tencent AI Lab

- Problem Definition
- Representation Learning on Graph Structure
- Representation Learning on Attributed Graph

Problem Definition

Tencent AI Lab

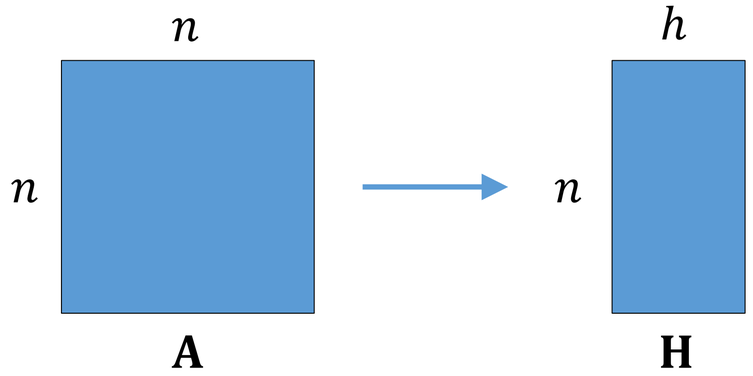
- What is the representation learning on graphs?



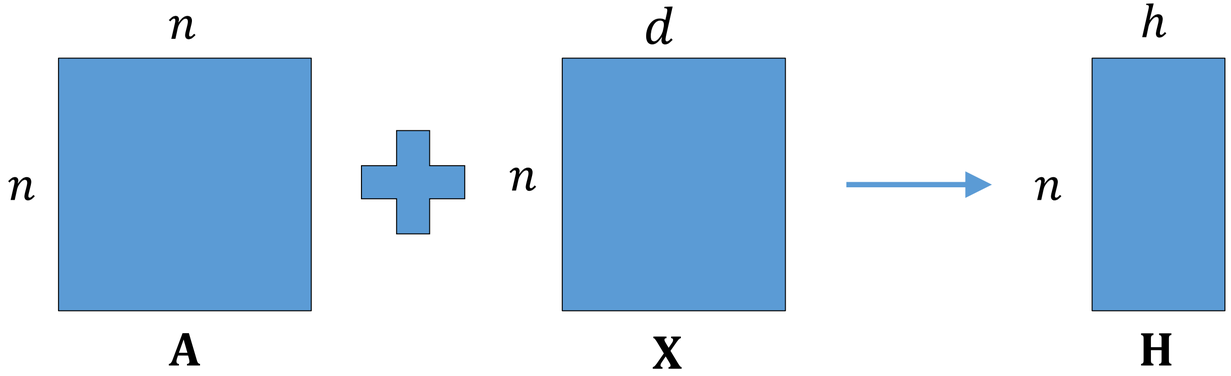
- Why we want to do this on graphs?
  - Graphs usually represented as adjacency list/matrix.
  - Machine Learning algorithms are not friendly to such adjacency representations.
  - Continuous valued vectors are much more model friendly.
  - Can apply classical ML methods.
- **Our Goal:** Conduct the graph inference & analytics in vector space.

Two perspectives

- From graph theory and data mining perspective: **how to represent/preserve the graph topology in vector space. (Representation Learning on Graph Structure)**
  - A is adjacency matrix. $d \ll n$



- From machine learning perspective: **how to generate the new features with the help of topological information of data points. (Representation Learning on Attributed Graph)**
  - W is constructed from graph topology.



Tencent
AI Lab

Representation Learning on Graph Structure

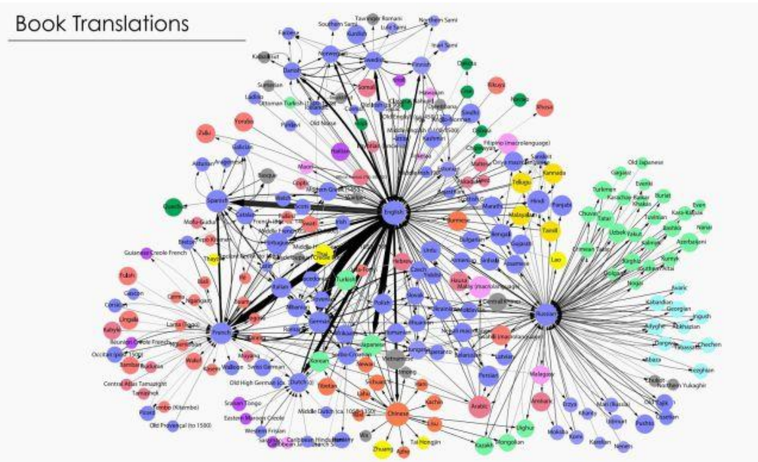
Representation Learning on Graph Structure

Tencent AI Lab

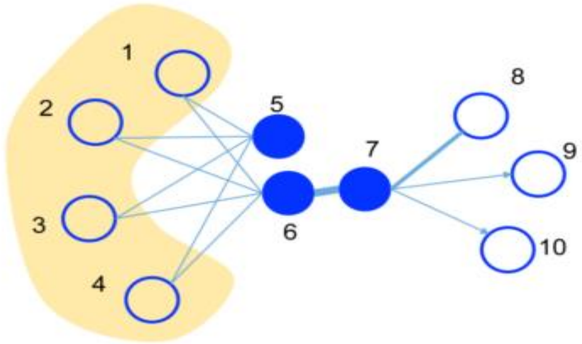
- Naïve Approach : Since we need: $n \times n \rightarrow n \times d$, $\mathbf{Y}$ can be naturally constructed by classical dimension reduction techniques.
  - High complexity.
  - Can not represent the topological relation between vertexes.

- **Preserve graph topology → Model the proximity between vertexes.**
- There are many ways to define the graph proximities.

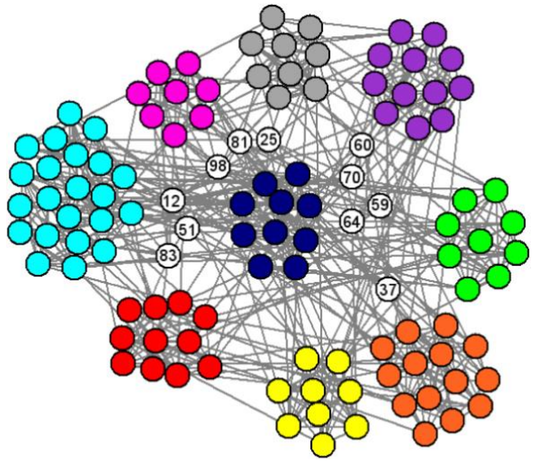
Co-occurrence (neighborhood)



High-order proximities



Communities

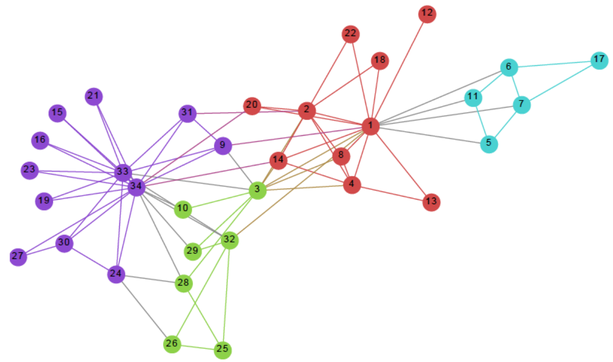


Different algorithms try to preserve different kinds of proximities in the vector space.

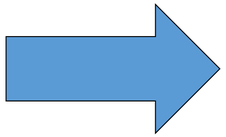
Modeling Vertex neighborhood

Tencent AI Lab

- The vertex neighborhood relation is important in networks.
- The vertex representation should be able to reflect its neighborhood information.
- How to define neighborhood in graphs?



- Exploit truncated random walk to define neighborhood of a vertex.



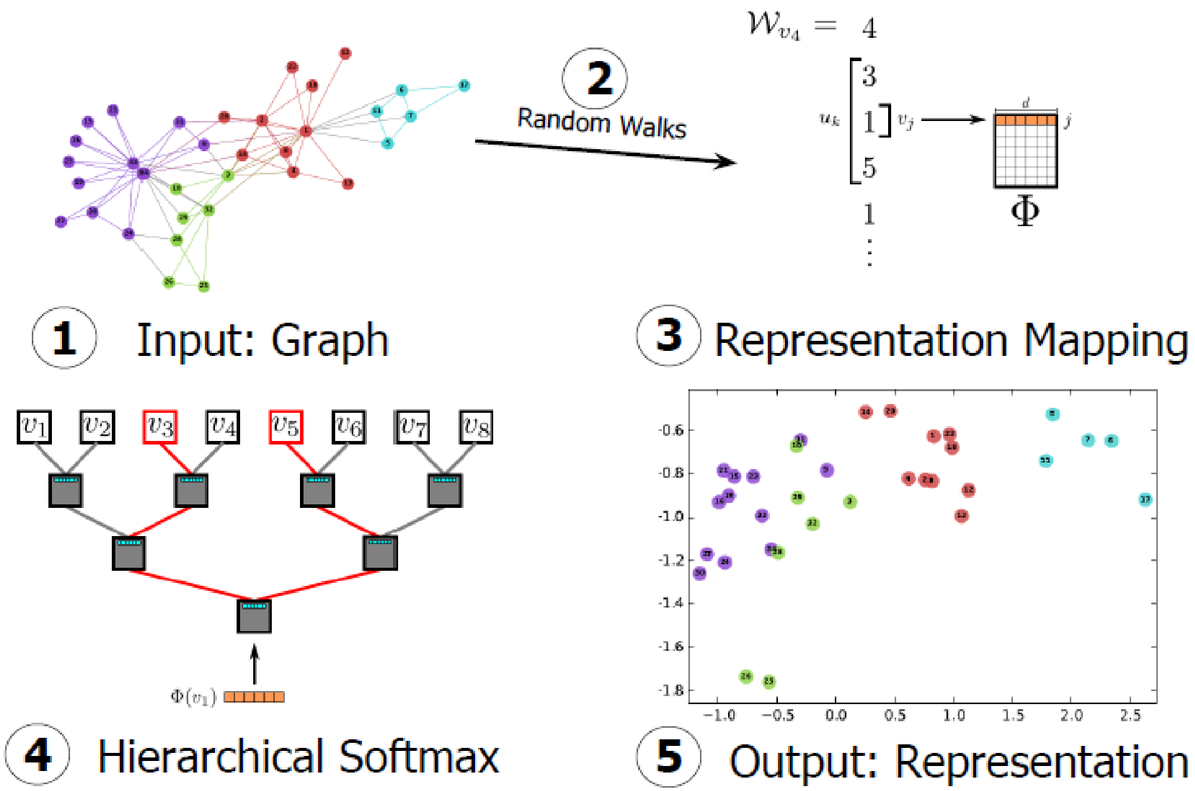
Random Walks on Graph

- $V_{26} - V_{25} - V_{32} - V_3 - V_{10} \cdots$
- $V_5 - V_7 - V_{17} - V_6 - V_{11} \ldots$
- $V_{31} - V_{33} - V_{21} - V_{33} - V_{15}$

Deepwalk

Tencent AI Lab

- Framework



- Core idea
  - Sample the truncated random walk path to represent the vertex neighborhood.
  - Treat the vertex path as “sentences” and employ the NLP model (word2vec) to learn the vertex embedding.
- Extension: many xxx2vec papers. ☺

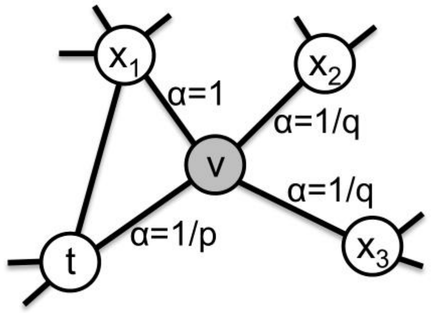
Extensions of Deepwalk

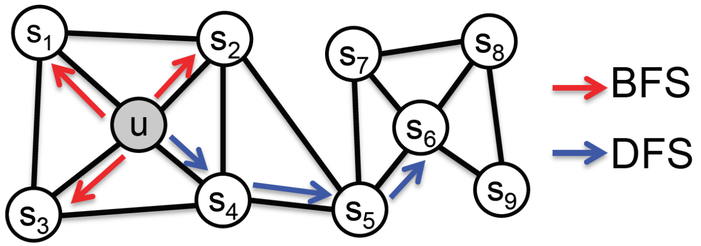
Tencent AI Lab

- **Node2vec**
  - Motivation: The truncated random walk is not good enough.
  - Core idea: replace the truncated random walk with a biased random walk.

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$

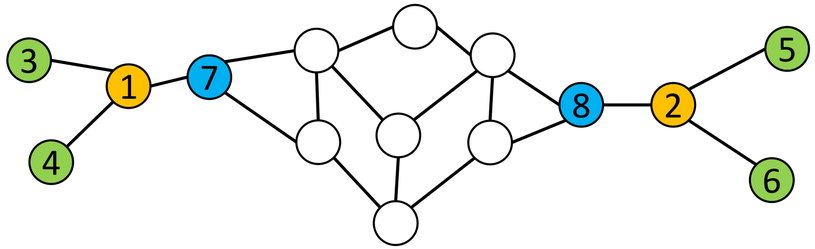
Smaller p: BFS
Smaller q: DFS



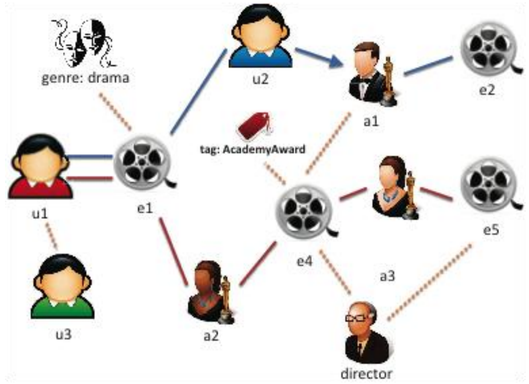


Homophily (community): DFS search
Structural equivalence: BFS search

- **Struc2vec**
  - Motivation: Capture the structural identity.
  - Core idea: Establish the new graph based on the structural similarity.



- **Metapath2vec**
  - Motivation: Different kinds of Graph type. (Heterogenous Graph)
  - Core idea: Random walk with predefined meta-path.
    e.g. author – conference – paper – author



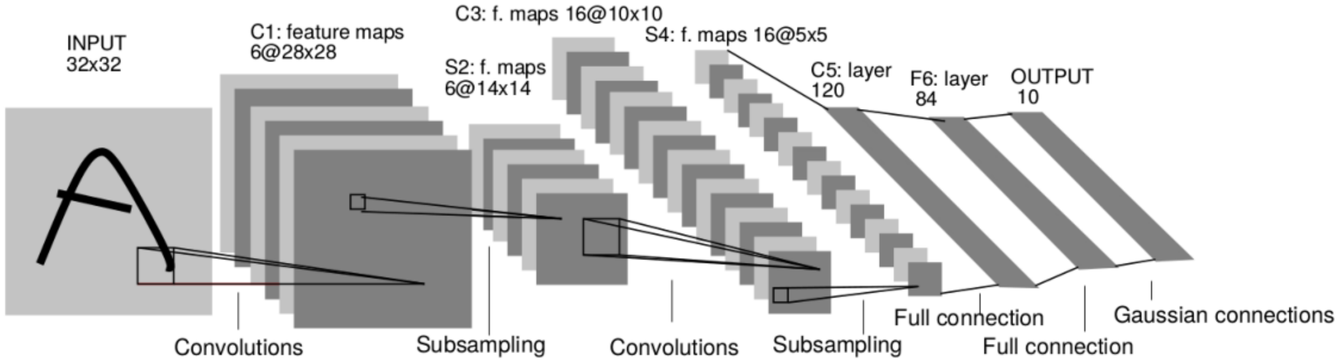
Tencent
AI Lab

Representation Learning on Attributed Graph

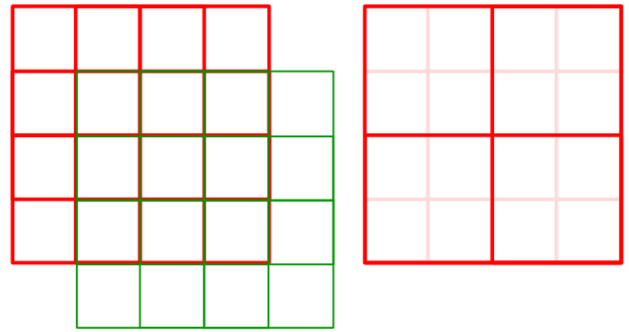
Representation Learning on Attributed Graph

Tencent AI Lab

- Instead of preserve topology, machine learning people is interested in generate the new vertex feature based on the graph topology.
- What we have now: Convolutional Neural Network -- A powerful and success representation learning tool.

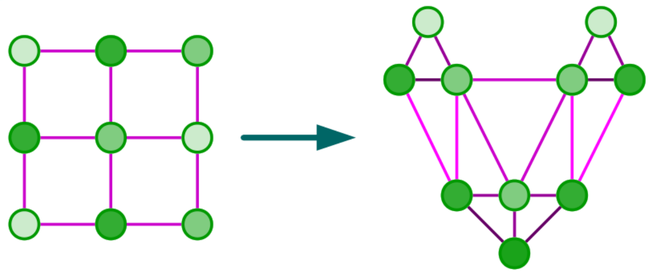


- View CNN as Euclidean domains or Grids:
  - Translation Invariance (yielding convolutions).
  - Multiscale structure (yielding downsampling).
  - Inductive bias that exploits stationarity and deformation stability of many tasks.
- Roadmap: extend CNNs to non-Euclidean geometries by replacing filtering and pooling by appropriate operators.



Representation Learning on Attributed Graph

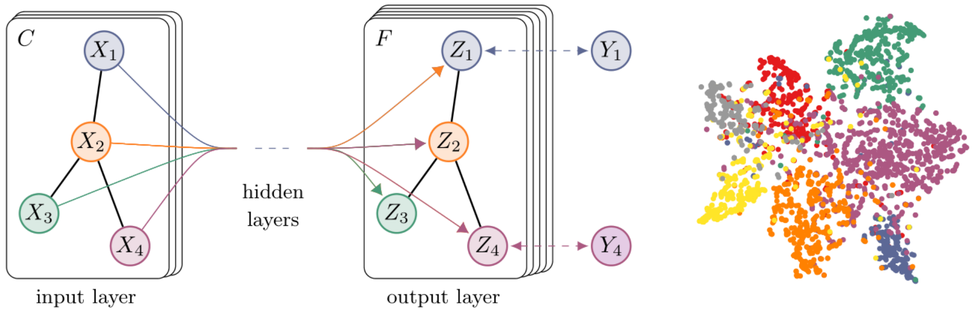
Tencent AI Lab



- Graphs vs Euclidean grids:
  - Irregular sampling.
  - Weighted edges.
  - No orientation or ordering (in general).
- **Challenge:** efficient formulation of convolution and down-sampling on graphs.
- **Graph Convolutional Network (GCN)**

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

$\tilde{A} = A + I_N$: Adjacency matrix with self-connections.
$\tilde{D} = \sum_j \tilde{A}_{ij}$: Degree matrix.
H: Feature map.
W: Trainable parameters.



(a) Graph Convolutional Network

(b) Hidden layer activations

- **The supervised loss:**

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

$$\mathcal{L} = -\sum_{l\in\mathcal{Y}_L}\sum_{f=1}^{F} Y_{lf}\ln Z_{lf} ,$$

GCN: From Spectral to Spatial

Tencent AI Lab

- Consider spectral convolutions on graphs defined as the multiplication of a signal (a scalar for every node) with a filter.

$$y = x *_{\mathcal{G}} g = U \begin{bmatrix} \hat{g}(\lambda_1) & & 0 \\ & \ddots & \\ 0 & & \hat{g}(\lambda_n) \end{bmatrix} U^T x = U\hat{g}(\Lambda)U^T x = \hat{g}(L)x$$

U: the matrix of eigenvectors of the normalized graph Laplacian $L = I_N - D^{\frac{1}{2}}AD^{\frac{1}{2}} = U\Lambda U^T$

U is the graph Fourier basis, Λ are graph `frequencies`.

- **Parametrization:** replace $\hat{g}(\Lambda)$ to $\hat{g}_\theta = diag(\theta)$
  - Still computationally expensive (multiplication with the eigenvector matrix U is $\mathcal{O}(N^2)$).
- **Polynomial parametrization by Chebyshev polynomials**

$$y = \hat{g}_\theta(L)x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{L})x = \sum_{k=0}^{K-1} \theta_k \bar{x}_k, \quad \tilde{L} = \frac{2}{\lambda_n}L - I_n$$

When K = 1 (only consider the neighborhood), we have:

$$g_\theta \star x \approx \theta \left(I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}\right) x$$

Spectral **Param. No.**

$n \times |g|$

n

K

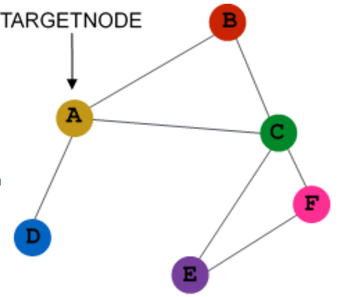
Spatial

1

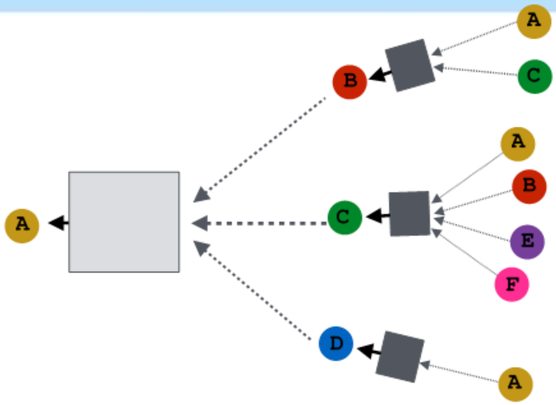
GCN: Extension

Tencent AI Lab

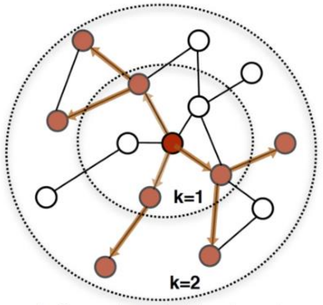
- Recall: $$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right).$$



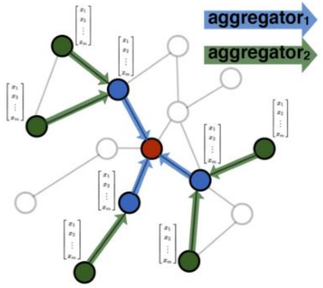
INPUT GRAPH



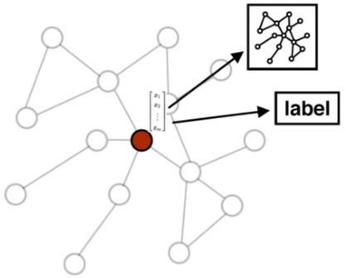
- Another simpler perspective: **GCN is the aggregation of neighbor features.**
- GraphSAGE: simple aggregation is not enough.
  - Inductive Learning Setting vs Transductive Learning Setting.
  - Provide the la ion).



1. Sample neighborhood

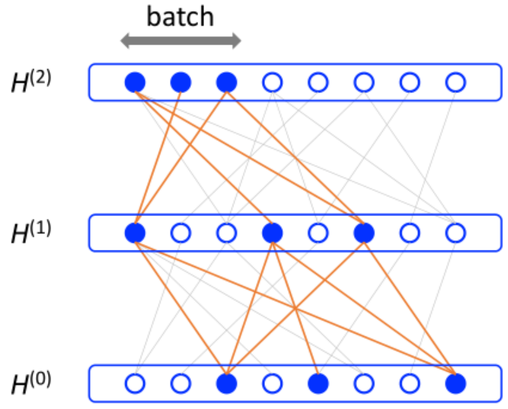


2. Aggregate feature information from neighbors



3. Predict graph context and label using aggregated information

- FastGCN: aggregate all neighbor is not efficient.
  - Sample fixed number of vertex per layer with bottom-up style.
  - The sampler is designed for variance reduction.
  - Avoid the explosion of expanded neighborhood.

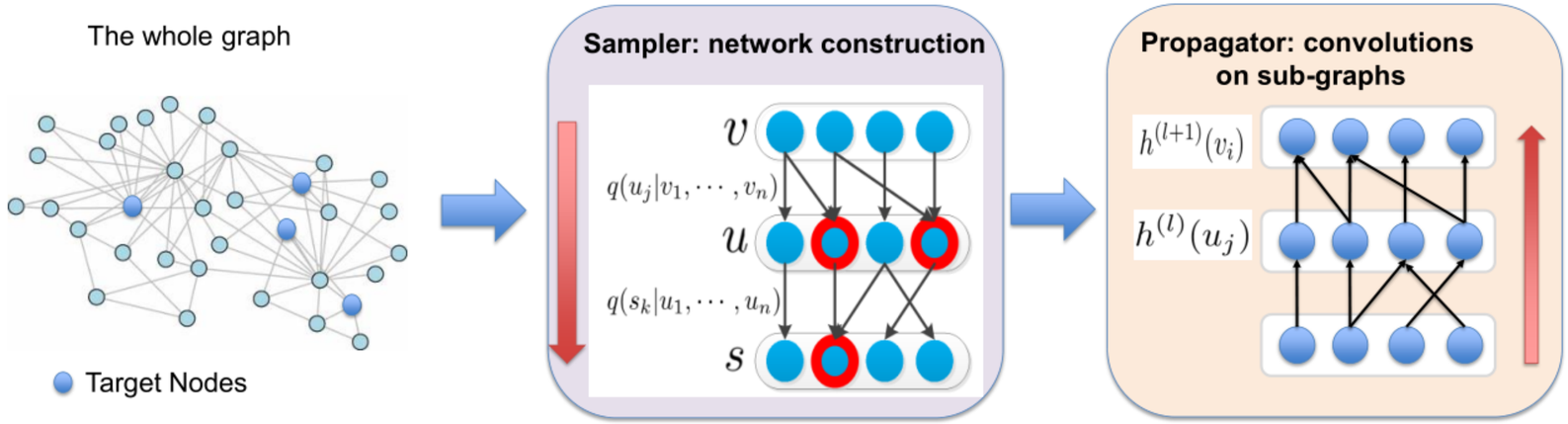


Graph convolution view

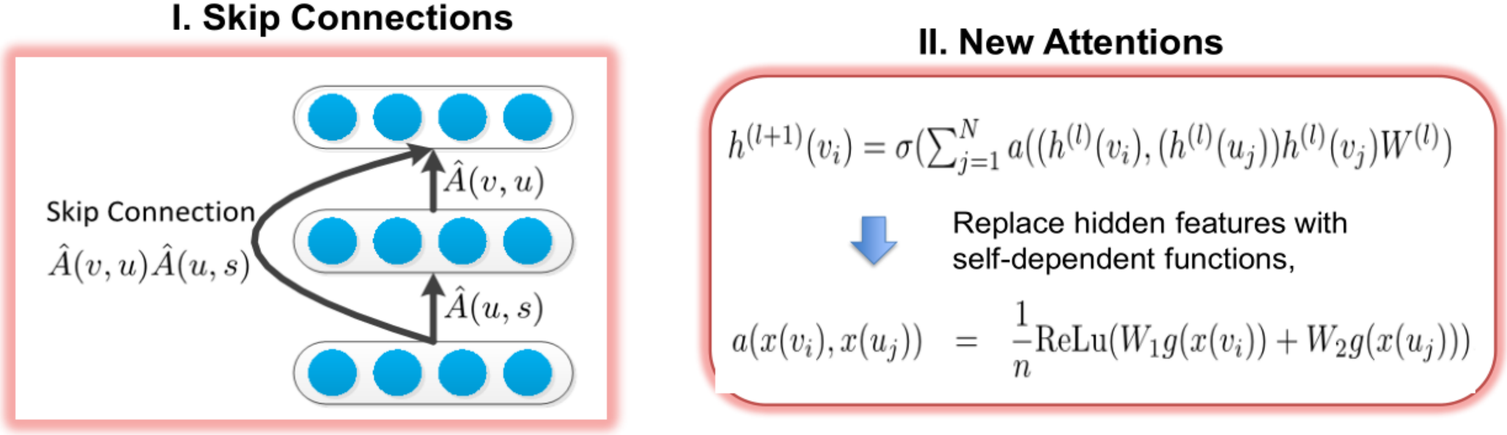
GCN: Extension

Tencent AI Lab

- AS-GCN: A more efficient sample strategy
  - Top-down sampler, ensure the connection between two layers are dense.

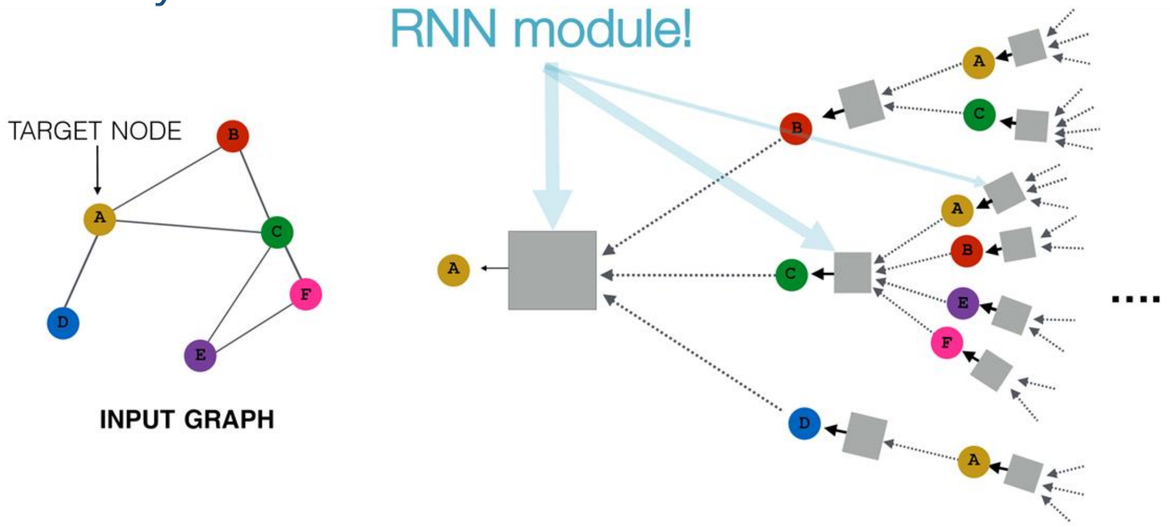


  - Two more things: skip connection and attentions

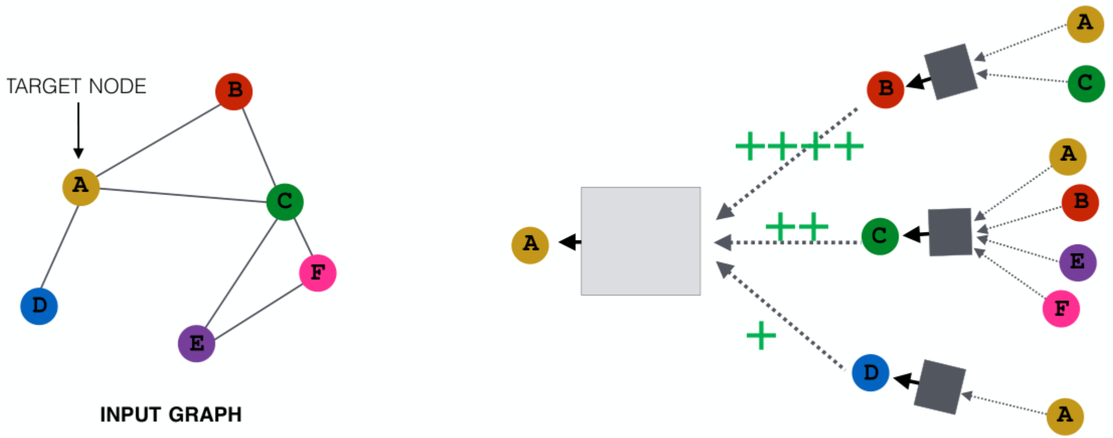


GCN: Extension

- Gated Graph Neural Networks: State and Message Passing
  - Parameter sharing across layers.

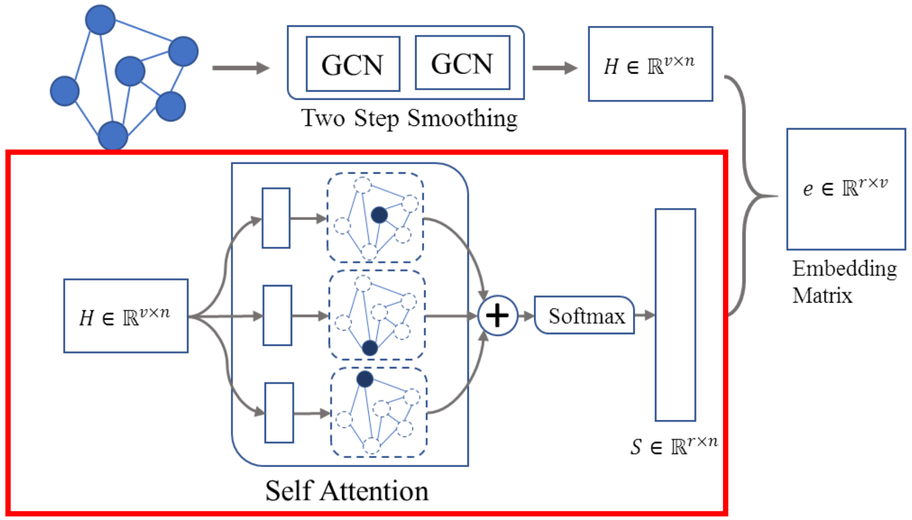


- Graph Attention Networks.

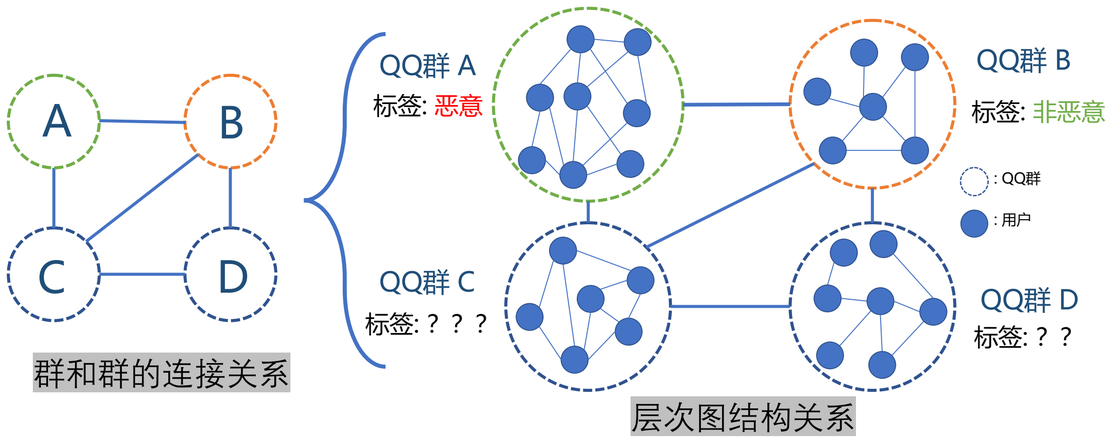


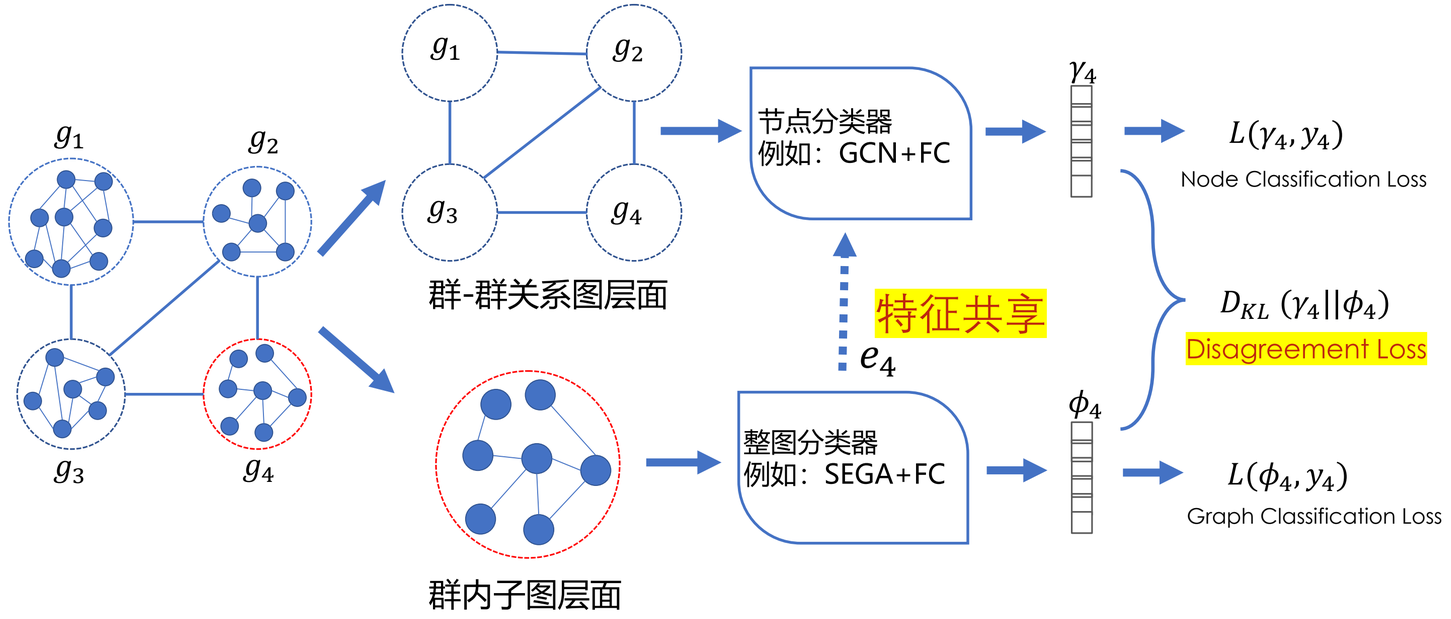
GCN: Extension

- SEAL: Towards complex graph structures.
- Graph-based Representation Learning.



- Combine node-level and graph-level optimization goals.





Take-away

Tencent AI Lab

- According to random walk technique, the graph representation problem can be converted to a NLP-like problem. A huge amount of methods are based on this conversion.
- Graph proximity is a very useful measurement in the unsupervised graph representation learning. A huge amount of papers are based on the redefinition of this measurement.
- Sampling and attention are two very powerful techniques when we build graph representation models.

Bibliography

Tencent AI Lab

- [Deepwalk] Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." In KDD, 2014.
- [Node2vec] Grover, Aditya, and Jure Leskovec. "node2vec: Scalable feature learning for networks." In KDD, 2016.
- [Struc2vec] Ribeiro, Leonardo FR, Pedro HP Saverese, and Daniel R. Figueiredo. "struc2vec: Learning node representations from structural identity. " In KDD, 2017.
- [Metapath2vec] Dong, Yuxiao, Nitesh V. Chawla, and Ananthram Swami. "metapath2vec: Scalable representation learning for heterogeneous networks." In KDD, 2017.
- [LINE] Tang, Jian, et al. "Line: Large-scale information network embedding." In WWW, 2015.
- [SDNE] Wang, Daixin, Peng Cui, and Wenwu Zhu. "Structural deep network embedding." In KDD, 2016.
- [M-NMF] Wang, Xiao, et al. "Community Preserving Network Embedding." *AAAI*. 2017.
- [GCN] Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." arXiv preprint arXiv:1609.02907 (2016).
- [GraphSAGE] Hamilton, Will, Zhitao Ying, and Jure Leskovec. "Inductive representation learning on large graphs." In NIPS. 2017.

Bibliography

- [FastGCN] Chen, Jie, Tengfei Ma, and Cao Xiao. "FastGCN: fast learning with graph convolutional networks via importance sampling." *arXiv preprint arXiv:1801.10247* (2018).
- [AS-GCN] Huang, Wenbing, et al. "Adaptive Sampling Towards Fast Graph Representation Learning." In NIPS 2018.
- [SEAL]Jia Li, Yu Rong, Hong Cheng, Helen Meng, Wenbing Huang, Junzhou Huang. " Semi-Supervised Graph Classification: A Hierarchical Graph Perspective " In WWW 2019.
- [GAT] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, Yoshua Bengio. Graph Attention Networks