Data Mining:

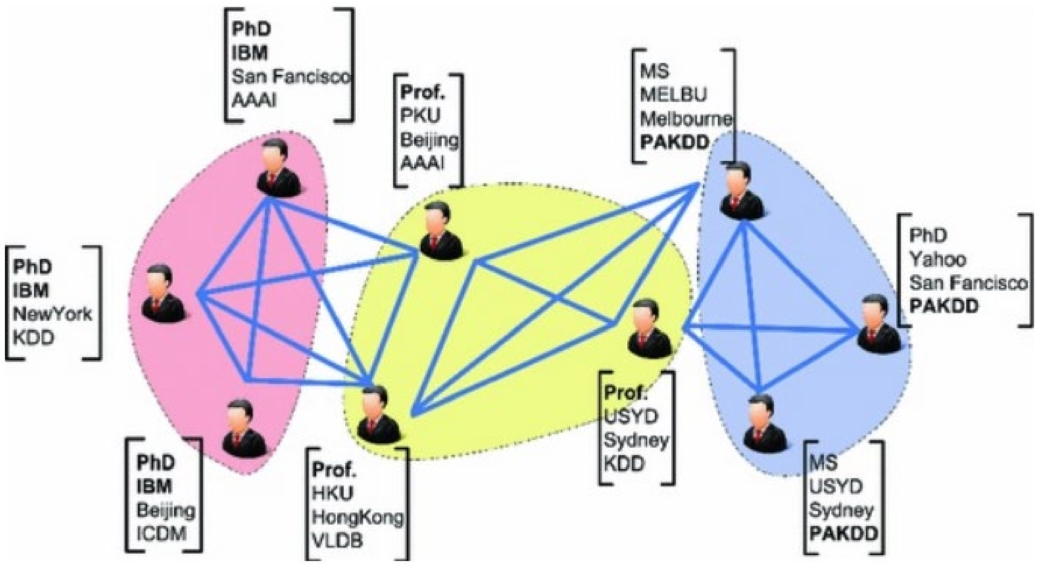
Advanced Techniques

Graph Mining: part 2

- 主讲教师: 陈佳伟, sleepyhunt@zju.edu.cn
  - https://jiawei-chen.github.io/
- TA: 陈思睿, chenthree@zju.edu.cn

Review --- Graph data

- A graph is a mathematical structure used to model pairwise relations between objects.
- Graph can be formally defined as: $G = \{V, E, X\}$
  - V denotes the set of nodes
  - E denotes the set of edges between nodes
  - X denotes the set of features of nodes

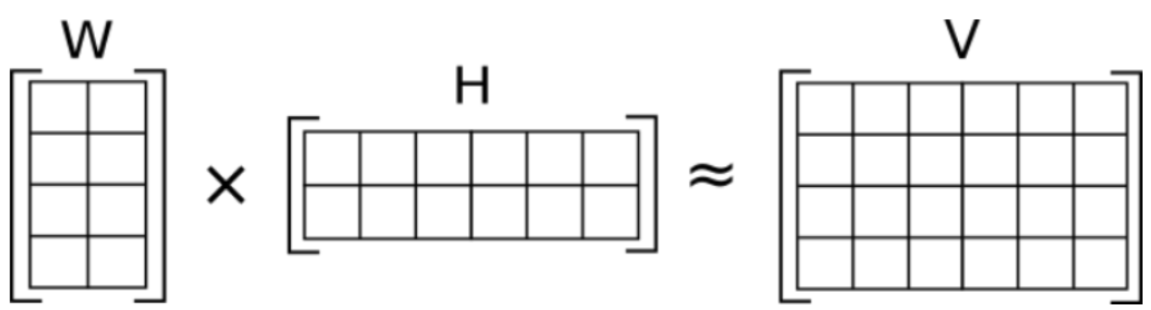


Review---Matrix factorization

- Matrix factorization
  - factorizing a matrix into a product of two lower-dimensional matrices
  - The goal is to approximate the original matrix by capturing its underlying structure and patterns

W, H: embedding of nodes

V: graph features



Slide from Sheng Zhou

Review---Random walk

- Node representation via random walk
  - Performing random walk on graph to generate multiple sequences of nodes
  - utilizing word2vec technique to get the node representation

Slide from Sheng Zhou

Review---finding new idea

掌握更多*的*方法：

关注其他领域的方法

深入理解问题：

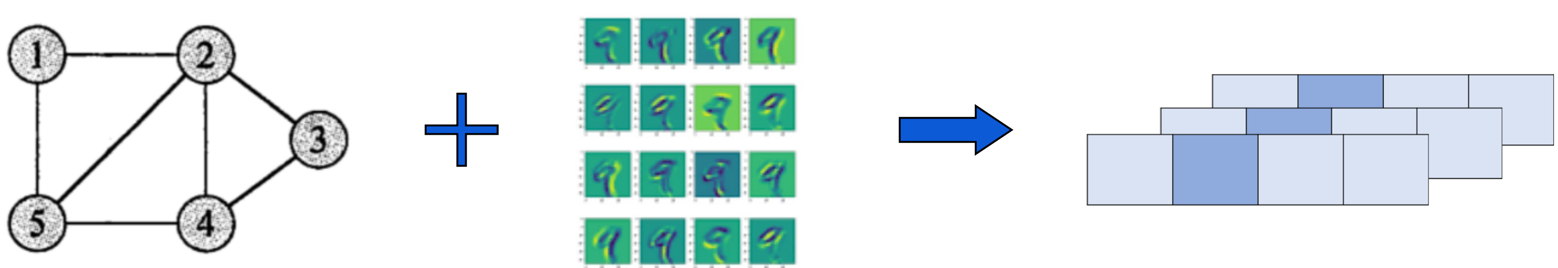
对该领域全面调研

Graph Neural Network

- Graph data
- Classic graph representation learning
- Graph neural network
  - Three perspectives
  - Applications
  - Promising directions

Motivations

- Weaknesses of classic methods
  - Node features have not been encoded
  - Graph structure have not been explicitly utilized
  - Completely unsupervised manner
    - Labels, if available, can not be utilized
- Desirable model: a “neural-network-style” model that explicitly encode features and graph structure $f(A, X) -> H$



Motivations

$$f(A, X) -> H$$

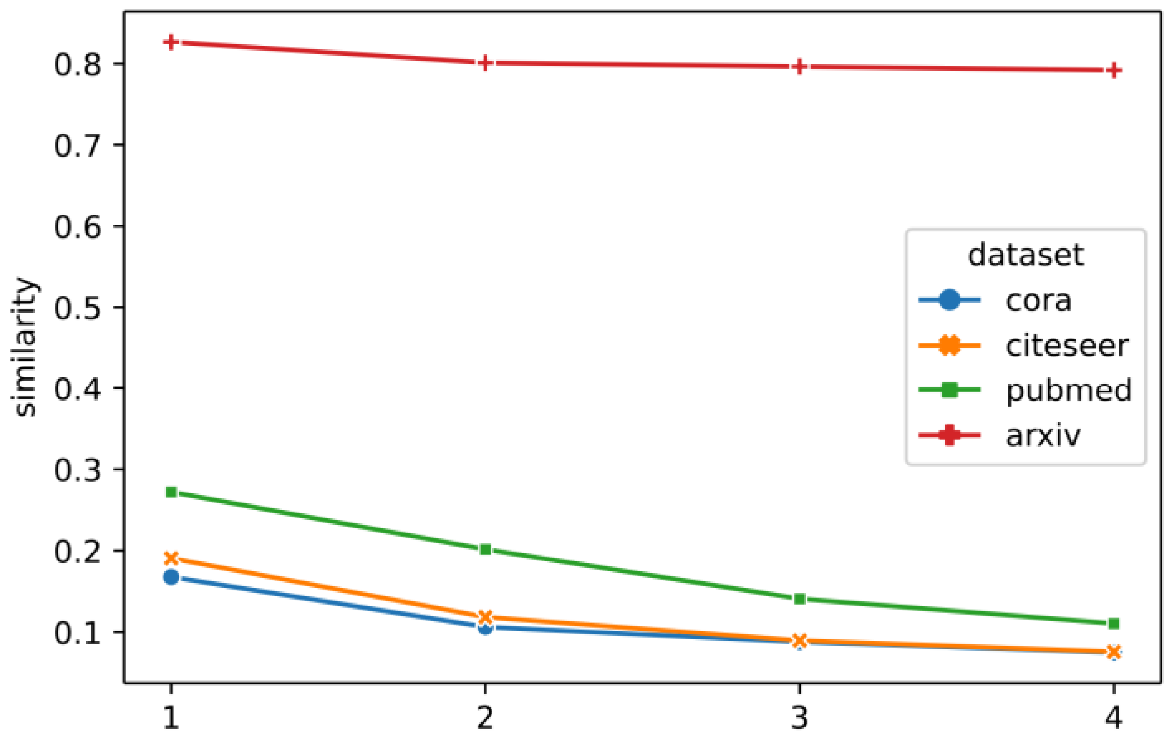
- Challenges:
  - The **sparsity** and **large-scale** of A
  - **Permutation Invariance**: the representation should not change when the order of nodes or vertices is permuted.

$$f\left(\mathbf{P}\mathbf{A}\mathbf{P}^{\top}\right) = f(\mathbf{A})$$
$$f\left(\mathbf{P}\mathbf{A}\mathbf{P}^{\top}\right) = \mathbf{P}f(\mathbf{A})$$

- Conventional neural network does not meet permutation invariance.

Motivations

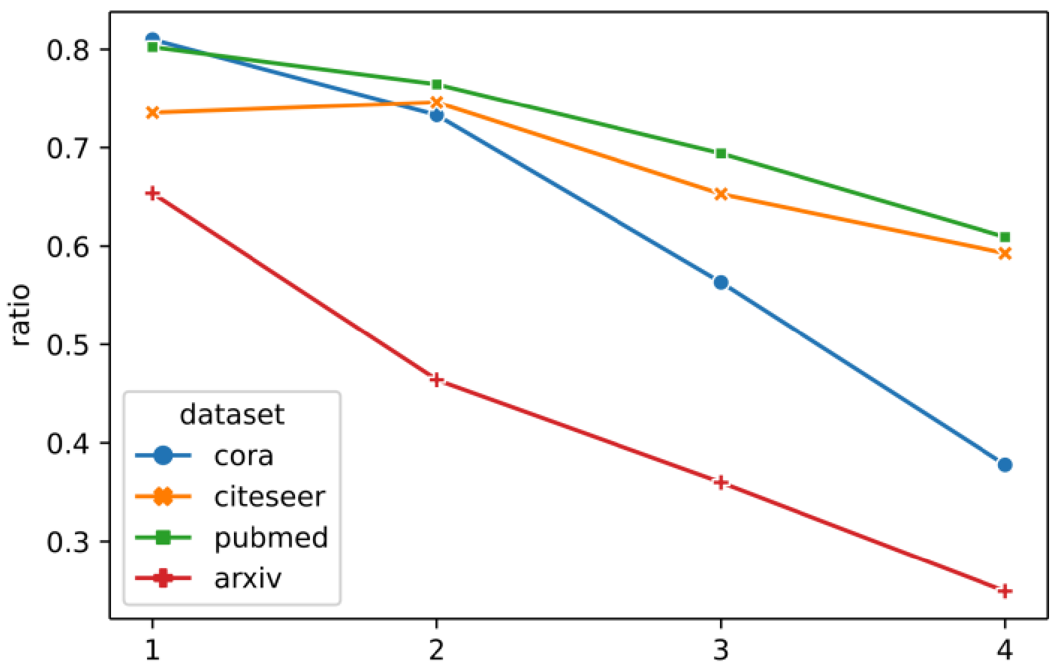
- Homophily (同质性) assumption:
  - Nodes with similar attributes or features are more likely to be connected or have edges between them.



Feature similarity with the graph distance

Motivations

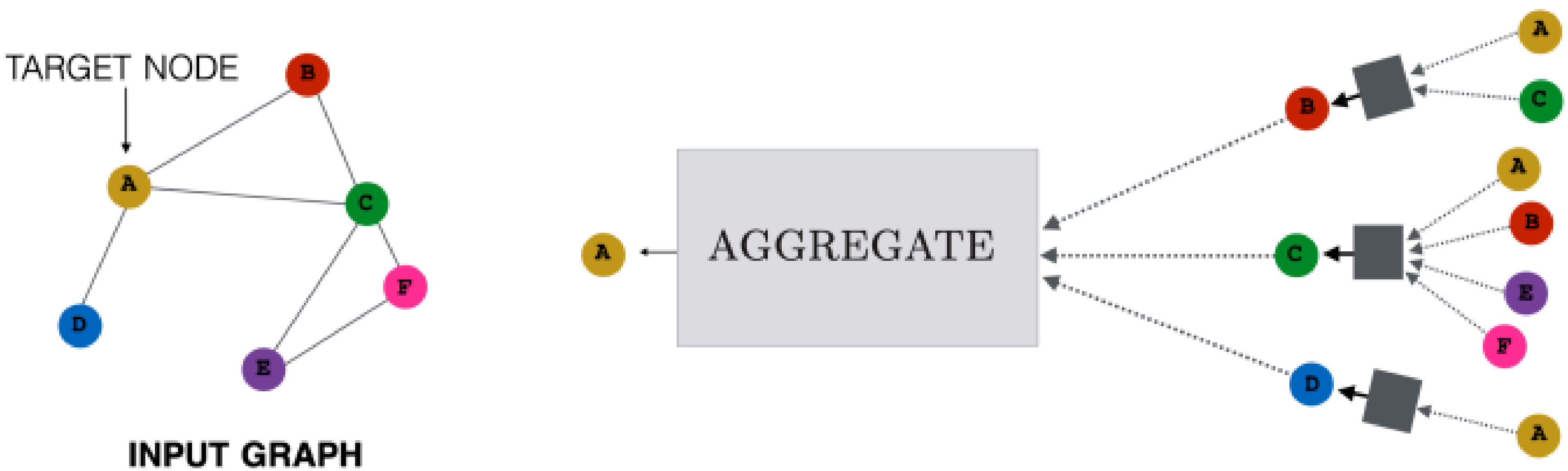
- Homophily (同质性) assumption:
  - Nodes with similar attributes or features are more likely to be connected or have edges between them.



Label similarity with the graph distance

GNN from Spatial Perspective

- Neural Message Passing
  - propagating and aggregating information between nodes in a graph.



Homophily assumption

GNN from Spatial Perspective

- Formal definition

$$\mathbf{h}_u^{(k+1)} = \mathrm{UPDATE}^{(k)}\left(\mathbf{h}_u^{(k)},\ \text{AGGREGATE}^{(k)}\left(\{\mathbf{h}_v^{(k)}, \forall v \in \mathcal{N}(u)\}\right)\right)$$
$$= \mathrm{UPDATE}^{(k)}\left(\mathbf{h}_u^{(k)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)}\right),$$

$$\mathbf{h}_u^{(0)} = \mathbf{x}_u \qquad \mathbf{z}_u = \mathbf{h}_u^{(K)}, \forall u \in \mathcal{V}$$

- Two key steps:
  - Aggregate: how to aggregate information from neighbors
  - Update: how to update the node features based on the information from the neighbors

GNN from Spatial Perspective

- Basic model

$$\mathbf{h}_u^{(k)} = \sigma\left(\mathbf{W}_{\text{self}}^{(k)}\mathbf{h}_u^{(k-1)} + \mathbf{W}_{\text{neigh}}^{(k)}\sum_{v\in\mathcal{N}(u)}\mathbf{h}_v^{(k-1)} + \mathbf{b}^{(k)}\right)$$

  - Aggregate: directly average the information of neighbors

$$\mathbf{m}_{\mathcal{N}(u)} = \text{AGGREGATE}^{(k)}\left(\{\mathbf{h}_v^{(k)}, \forall v \in \mathcal{N}(u)\}\right)$$

  - Update: update the node feature via a linear layer:

$$\text{UPDATE}\left(\mathbf{h}_u, \mathbf{m}_{\mathcal{N}(u)}\right) = \sigma\left(\mathbf{W}_{\text{self}}\mathbf{h}_u + \mathbf{W}_{\text{neigh}}\mathbf{m}_{\mathcal{N}(u)}\right)$$

GNN from Spatial Perspective

- Generic models
  - Aggregation
    - Neighbor Pooling
    - Neighbor Normalization
    - Neighbor Attention
  - Update
    - Concatenation
    - Gated Update

GNN from Spatial Perspective

- Neighbor Pooling
  - Mapping a set of vectors from neighbors to a target vector
  - Neighbor Pooling maintains **permutation invariance** by treating neighboring nodes as a set
  - Mean pooling is the most commonly-used function

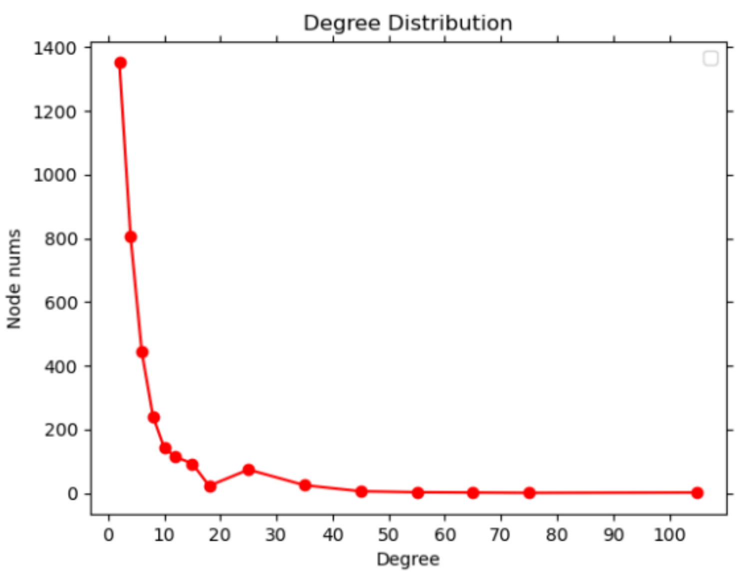
$$\mathbf{m}_{\mathcal{N}(u)} = \mathbf{MLP}_\theta\left(\sum_{v\in N(u)} \mathrm{MLP}_\phi\left(\mathbf{h}_v\right)\right)$$

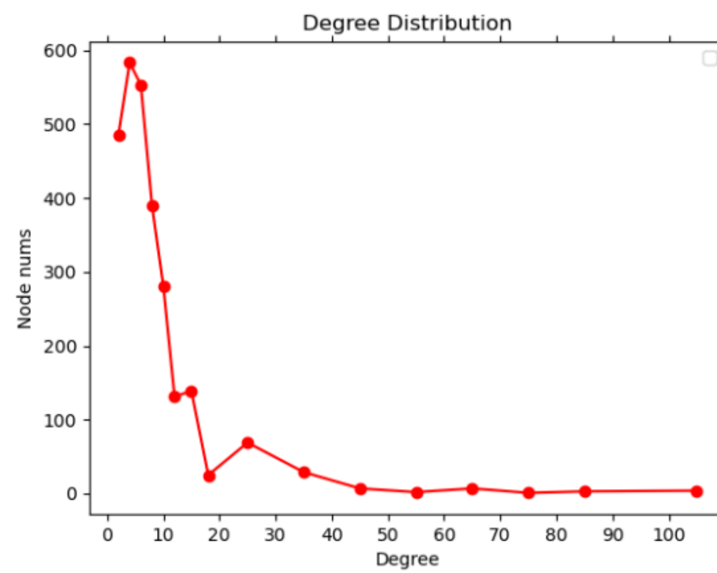
  - *This formula can represent any* ***set function invariant*** *to the permutation of neighbors*

NIPS'17: Deep sets.

GNN from Spatial Perspective

- Neighbor Normalization
  - Motivations: The long-tailed nature of nodes degrees
  - high-degree nodes could dominate the aggregation process
  - explosion of feature magnitude





GNN from Spatial Perspective

- Neighbor Normalization: two conventional strategies
  - Discounted by the degree of the target nodes

$$\mathbf{m}_{\mathcal{N}(u)} = \frac{\sum_{v \in \mathcal{N}(u)} \mathbf{h}_v}{|\mathcal{N}(u)|}$$

  - Considering the degrees of both the target node and the neighbor

$$\mathbf{m}_{\mathcal{N}(u)} = \sum_{v \in \mathcal{N}(u)} \frac{\mathbf{h}_v}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(v)|}}$$

sigir'22 : investigating Accuracy-Novelty Performance for Graph-based Collaborative Filtering

GNN from Spatial Perspective

- Neighbor Attention
  - The qualities of the information from different nodes are diverse
  - More similar nodes could be more useful
  - Introducing edge weights to capture the importance of neighbors

$$\mathbf{m}_{\mathcal{N}(u)} = \sum_{v \in \mathcal{N}(u)} \alpha_{u,v} \mathbf{h}_v$$

GNN from Spatial Perspective

- Graph attention network

$$\alpha_{u,v} = \frac{\exp\left(\mathbf{a}^\top \left[\mathbf{W}\mathbf{h}_u \oplus \mathbf{W}\mathbf{h}_v\right]\right)}{\sum_{v' \in \mathcal{N}(u)} \exp\left(\mathbf{a}^\top \left[\mathbf{W}\mathbf{h}_u \oplus \mathbf{W}\mathbf{h}_{v'}\right]\right)}$$

- Other forms of attention

$$\alpha_{u,v} = \frac{\exp\left(\mathbf{h}_u^\top \mathbf{W}\mathbf{h}_v\right)}{\sum_{v' \in \mathcal{N}(u)} \exp\left(\mathbf{h}_u^\top \mathbf{W}\mathbf{h}_{v'}\right)}$$

$$\alpha_{u,v} = \frac{\exp\left(\mathrm{MLP}\left(\mathbf{h}_u, \mathbf{h}_v\right)\right)}{\sum_{v' \in \mathcal{N}(u)} \exp\left(\mathrm{MLP}\left(\mathbf{h}_u, \mathbf{h}_{v'}\right)\right)}$$

ICLR’18 : graph attention network

GNN from Spatial Perspective

- Summary of neighbor aggregation
  - Neighbor Pooling
  - Neighbor Normalization
  - Neighbor Attention
- The key lies on maintain permutation invariant
- Which one is better?
  - According to the specific task and data
  - “黑猫白猫，能抓老鼠的就是好猫”
  - Aggregation for complex graph, uncertainty, HIN, dynamic, etc.

GNN from Spatial Perspective

- Update function
  - Integrate the information from the neighbors to update the node embedding

$$\mathbf{h}_u = \mathrm{Update}(\mathbf{h}_u, \mathbf{m}_{N(u)})$$

  - Basic strategies: concatenation and linear combination
  - Some complex strategies target at mitigating over-smoothing
- **Definition of Oversmoothing.** The nodes become more similar with the layer of GNN increasing.
  - Nodes become indistinguishable.
  - The negligible effect of node features.

GNN from Spatial Perspective

- A toy example of over-smoothing

$$\mathrm{h}_u^{(t)} = W_1 \mathrm{h}_u^{(t-1)} + W_2 \frac{1}{d_u} \sum_{v \in N_u} \mathrm{h}_v^{(t-1)} + b$$

$$H^{(t)} = W_1 H^{(t-1)} + W_2 \tilde{A} H^{(t-1)} + b$$

  - We have:

$$H \rightarrow (I - W_1 - W_2 \tilde{A})^{-1} b$$

    - regardless of the input features

Arxiv'23 : A survey on oversmoothing in graph neural networks

GNN from Spatial Perspective

- **Concatenate** : preserving the inherent information of nodes during the update to mitigate the over-smoothing

$$\text{UPDATE}_{\text{concat}}\left(\mathbf{h}_u, \mathbf{m}_{\mathcal{N}(u)}\right) = \left[\text{UPDATE}_{\text{base}}\left(\mathbf{h}_u, \mathbf{m}_{\mathcal{N}(u)}\right) \oplus \mathbf{h}_u\right]$$

  - Directly concatenate the features of the node itself

GNN from Spatial Perspective

- **LSTM, GRU** : Memorize the long-term historical information of the node feature

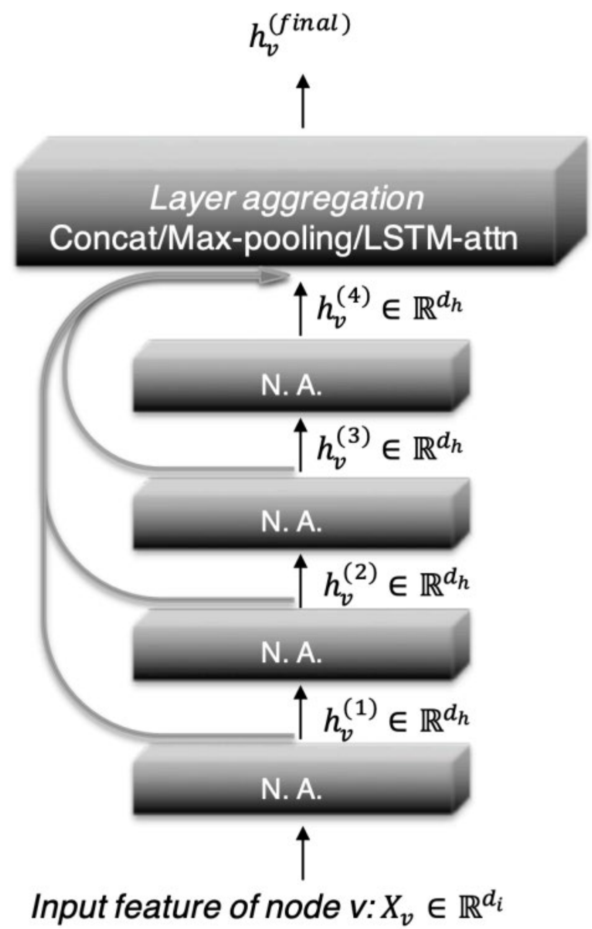
$$\mathbf{h}_u^{(k)} = \mathrm{GRU}\left(\mathbf{h}_u^{(k-1)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)}\right)$$

  - It can also be replaced by other RNN-type module.

GNN from Spatial Perspective

- **Jumping Knowledge Connections** : utilizing the information from all layers to get the final embeddings

$$\mathbf{z}_u = f_{\mathrm{JK}}\left(\mathbf{h}_u^{(0)} \oplus \mathbf{h}_u^{(1)} \oplus \ldots \oplus \mathbf{h}_u^{(K)}\right)$$



JKNet

GNN from Spatial Perspective

- How to train a GNN model?
  - Semi-supervised learning:

$$\mathcal{L} = \sum_{u \in \mathcal{V}_{\text{train}}} -\log\left(\operatorname{softmax}\left(\mathbf{z}_u, \mathbf{y}_u\right)\right)$$

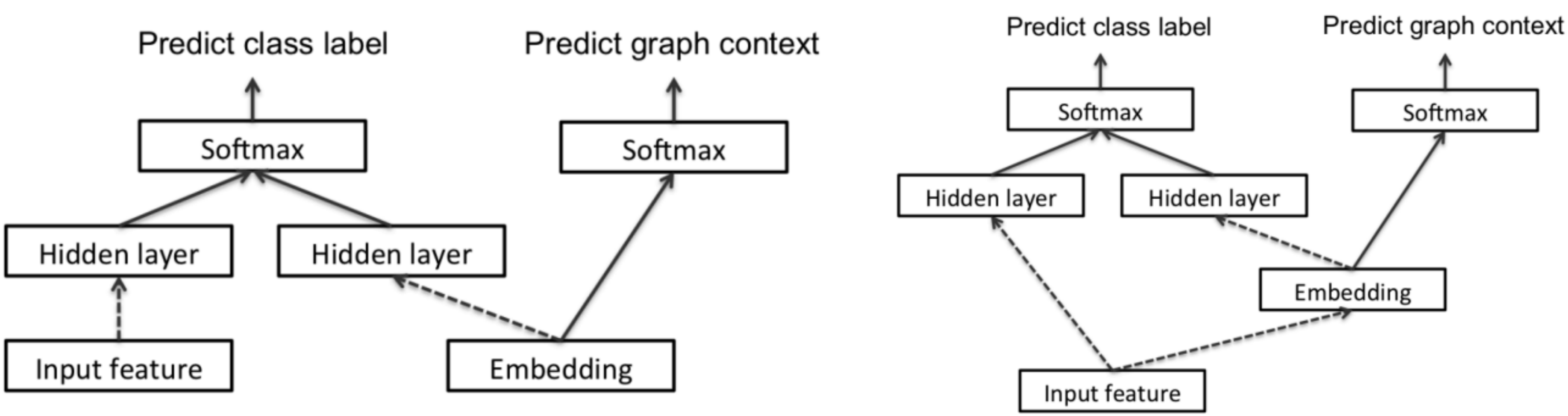
$$\operatorname{softmax}\left(\mathbf{z}_u, \mathbf{y}_u\right) = \sum_{i=1}^{c} \mathbf{y}_u[i] \frac{e^{\mathbf{z}_u^\top \mathbf{w}_i}}{\sum_{j=1}^{c} e^{\mathbf{z}_u^\top \mathbf{w}_j}}$$

  - Unsupervised (self-supervised) learning:
    - Formulate as a link prediction task

$$J_{\mathcal{G}}(z_u) = -\log(\sigma(z_u^T z_v)) - Q \cdot \mathbb{E}_{v_n \sim P_n(v)} \log(\sigma(-z_u^T z_{v_n}))$$

GNN from Spatial Perspective

- Transductive VS. Inductive ?



(a) Transductive Formulation

(b) Inductive Formulation

- Lies on how to generate embeddings in the test set.

Graph Neural Network

....

- Graph neural network
  - Three perspectives
    - Spatial Perspective
    - Spectral Perspective
    - Loss Perspective
  - Applications
  - Promising directions

GNN from Spectral Perspective

- Graph convolution network?
  - The relation with graph neural network?
- What is convolution （卷积）？

$$(f * g)(n) = \int_{-\infty}^{\infty} f(\tau)g(n-\tau)d\tau$$

  - 卷：invert the function g from $g(\tau)$ to $g(n-\tau)$
  - 积：cumulate the value of $f(\tau)g(n-\tau)$

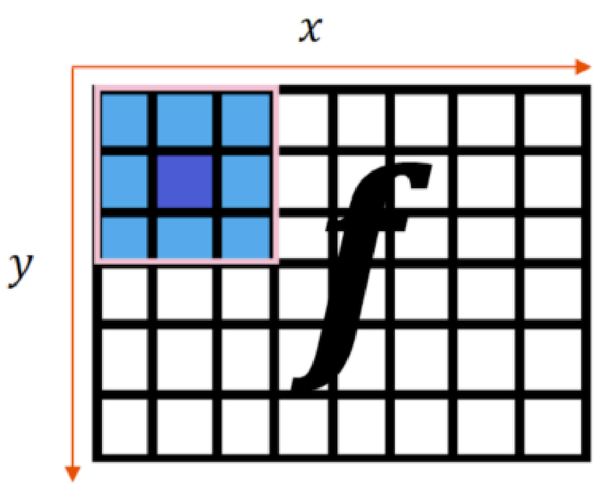
GNN from Spectral Perspective

- Convolutional network in computer vision
  - Motivations
    - **Translation invariance (平移不变性)**：the performance of the vision model does not change when the position of object is moved
    - **Locality (局部性)**： pixels that are close to each other are more likely to be semantically related or part of the same object than pixels that are far apart
    - **Efficiency & few parameters:** the large scale of the pixels

GNN from Spectral Perspective

- Convolutional network in computer vision

$$h(x,y) = (f * g)(x,y) = \sum_{m,n} f(x-m, y-n)g(m,n)$$



g=

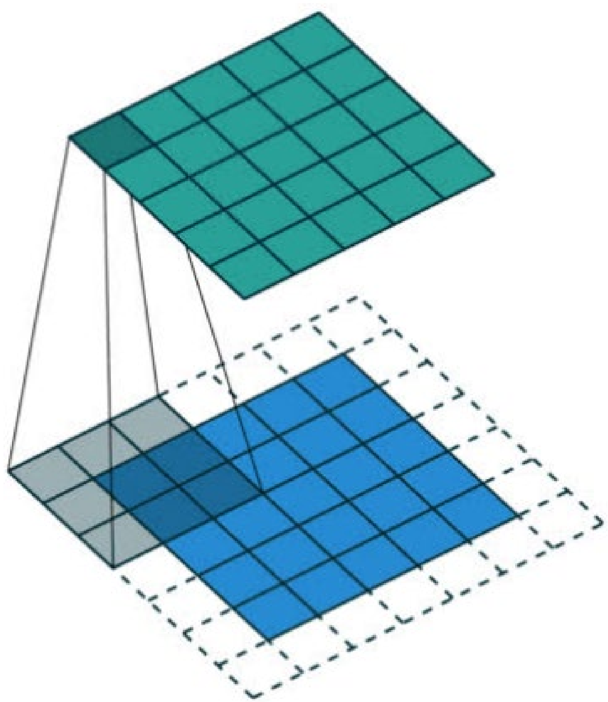
$g(1,1)$	$g(0,1)$	$g(-1,1)$
$g(1,0)$	$g(0,0)$	$g(-1,0)$
$g(1,-1)$	$g(0,-1)$	$g(-1,-1)$

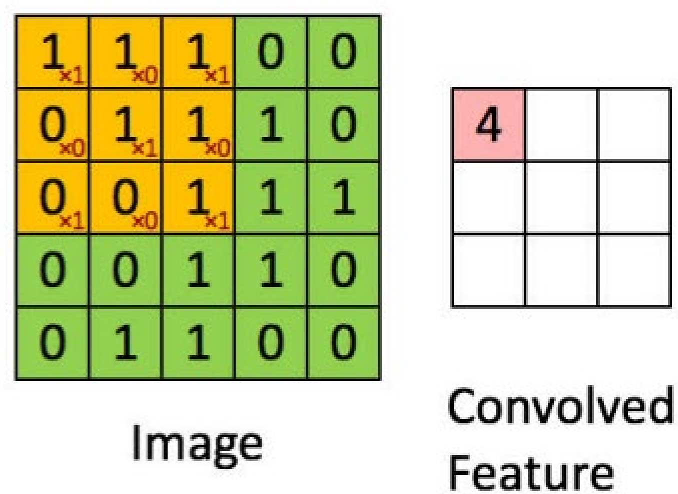
$$\begin{aligned} h(1,1) =& f(0,0)g(1,1) + f(1,0)g(0,1) + f(2,0)g(-1,1) \\ &+ f(0,1)g(1,0) + f(1,1)g(0,0) + f(2,1)g(-1,0) \\ &+ f(0,2)g(1,-1) + f(1,2)g(0,-1) + f(2,2)g(-1,-1) \end{aligned}$$

GNN from Spectral Perspective

- Convolutional network in computer vision

$$h(x,y) = (f * g)(x,y) = \sum_{m,n} f(x-m, y-n)g(m,n)$$





GNN from Spectral Perspective

- Transfer convolutional network to graph

Locality	→	**Homophily, Local structure**
Translation invariance	→	**Generalization**
Efficiency & Low Memory	→	**Large-scale & sparse graph**

GNN from Spectral Perspective

- Challenges:

Grid-like network

Irregular network

  - Permutation invariance
  - Undefination of f(x,**y-1**)?
  - Varying number of neighbors
    - hard to define convolution kernel

GNN from Spectral Perspective

- **Fourier Transform**: a linear integral transformation that converts **time-domain signals** into **frequency-domain signals**

GNN from Spectral Perspective

- **Fourier Transform**: converts **time-domain signals** $f(t)$ into **frequency-domain signals** $F(\mu)$.

$$F(\mu) = \int_{-\infty}^{+\infty} f(t) e^{-i2\pi\mu t} dt$$

$$f(t) = \int_{-\infty}^{+\infty} F(\mu) e^{i2\pi\mu t} d\mu$$

- Express $f(t)$ as a combination of sine and cosine waves $e^{i2\pi\mu t}$, weighted by $F(\mu)$, where $e^{i2\pi\mu t}$ is the basis.

GNN from Spectral Perspective

- **Euler's formula**:

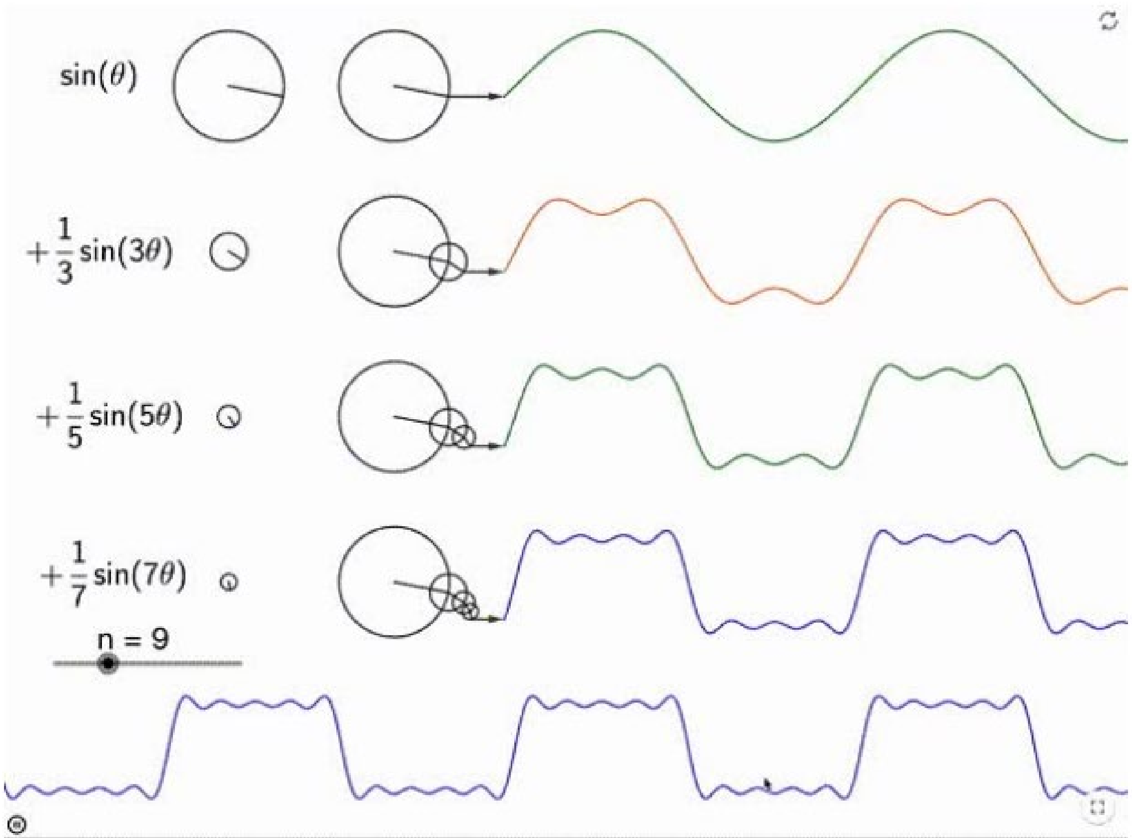
$$e^{i\theta} = i\sin\theta + \cos\theta$$

$$e^{i\pi} = -1$$

- Understanding the basis $e^{i2\pi\mu t}$
  - sine and cosine waves with different frequencies μ

GNN from Spectral Perspective

- sine and cosine waves:



信号的变换

Small μ
Smoothing

Large μ
oscillation

GNN from Spectral Perspective

- Advantages of **Fourier Transform**:
  - Disentangle the original signals according to the frequencies
    - **Different level of smoothing**
  - Operation on the frequency domain, e.g., filtering
  - Facilitate convolution
    - The **convolution** of two functions can be written as the inverse Fourier transform of the **product** of the Fourier transforms of these two functions.

$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}$$

GNN from Spectral Perspective

- Spectrum on graph
- The key lies on finding a proper space of basis
  - Indicating different levels of smoothing
  - Capturing graph structure
  - Orthogonal
  - Easy calculation

GNN from Spectral Perspective

- Laplace operator captures smoothing of a function
  - the divergence of the gradient of a scalar function
  - the Laplacian $\Delta f(x)$ of a function f at a point x measures by how much the average value of f over small balls centered at x deviates from $f(x)$.

$$\Delta f = \nabla^2 f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$$

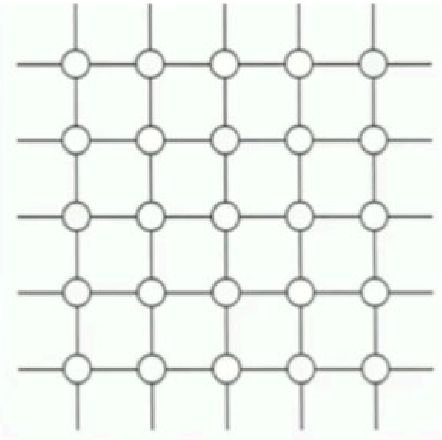
$$f(x+s) = f(\mathrm{x}) + s^T \nabla f + s^T H s + O(|s|^3)$$

GNN from Spectral Perspective

- Laplace operator on grid data

$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x)$$

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= f''(x) \approx f'(x) - f'(x-1) \\ &= f(x+1) + f(x-1) - 2f(x)\end{aligned}$$



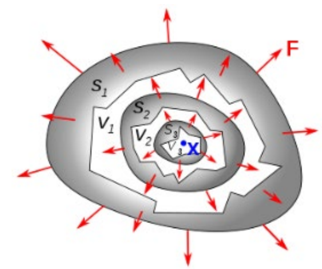
Grid-like network

$$\begin{aligned}\Delta f &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\ &= f(x+1,y) + f(x-1,y) - 2f(x,y) \\ &+ f(x,y+1) + f(x,y-1) - 2f(x,y) \\ &= f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)\end{aligned}$$

Differences between neighbors along four directions

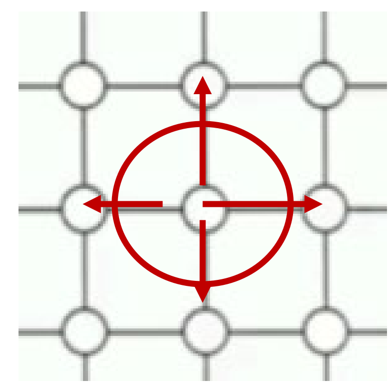
GNN from Spectral Perspective

- Laplace operator on grid data



$$\operatorname{div} \mathbf{F}|_{\mathbf{x}_0} = \lim_{V \to 0} \frac{1}{|V|} \oiint_{S(V)} \mathbf{F} \cdot \hat{\mathbf{n}}\, dS$$

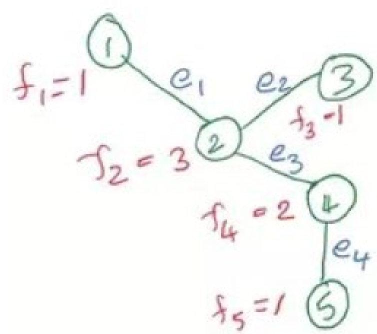
$$\Delta f(x,y) = div(\nabla f(x,y))$$
$$d_{\rightarrow} = f(x+1,y) - f(x,y)$$
$$d_{\leftarrow} = f(x-1,y) - f(x,y)$$
$$d_{\uparrow} = f(x,y+1) - f(x,y)$$
$$d_{\downarrow} = f(x,y-1) - f(x,y)$$
$$\Delta f(x,y) = d_{\rightarrow} + d_{\leftarrow} + d_{\uparrow} + d_{\downarrow}$$



GNN from Spectral Perspective

- Laplace operator on graph data
  - f: a function over nodes in graph
  - Gradient w.r.t edge: $\mathrm{grad}(f)|_e = f(u) - f(v)$
  - Define K as a $|V| \times |E|$ incidence matrix
  - $\mathrm{g} = \mathrm{grad}(f)|_E$

    $= K^T f$

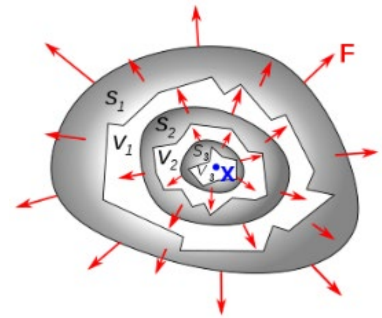


$$f = \begin{array}{c} \begin{bmatrix} 1 \\ 3 \\ 1 \\ 2 \\ 1 \end{bmatrix} \begin{array}{c} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{array} \\ 5\times 1 \end{array} \quad K = \begin{array}{c} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} \\ \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \end{array} \; 5\times 4$$

$$\mathrm{grad}(f) = K^T f = \begin{array}{c} \begin{bmatrix} -2 \\ 2 \\ 1 \\ 1 \end{bmatrix} \\ 4\times 1 \end{array} \begin{array}{l} e_1 : f_1 - f_2 \\ e_2 : f_2 - f_3 \\ e_3 : f_2 - f_4 \\ e_4 : f_4 - f_5 \end{array}$$

GNN from Spectral Perspective

- Laplace operator on graph data
  - Divergence at a point gives the net outward flux of a vector field
  - Divergence in graph as the collected flows over edges in a graph



$$g = \begin{array}{c} \begin{bmatrix} -2 \\ 2 \\ 1 \\ 1 \end{bmatrix} \\ 4\times 1 \end{array} \begin{array}{l} e_1 \\ e_2 \\ e_3 \\ e_4 \\ \end{array} \qquad \mathrm{div}(g) = Kg = \begin{array}{c} \begin{bmatrix} -2 \\ 5 \\ -2 \\ 0 \\ 1 \end{bmatrix} \\ 5\times 1 \end{array} \begin{array}{l} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ \end{array}$$

GNN from Spectral Perspective

- Laplace operator on graph data
  - $\Delta(f) = div(grad(f)) = KK^T f$
  - $L = KK^T$ named as a Laplacian matrix of graph
  - $L = D - A$?

$$KK^T = \underset{5\times 4}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}} \underset{4\times 5}{\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}} = \underset{5\times 5}{\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}} = L$$

GNN from Spectral Perspective

- Laplace operator on graph data
  - $L = D - A$? YES!
- Let $K = [e_1, e_2, \ldots . e_m]$

$$KK^T = \sum_{1 \le k \le m} e_k e_k^T$$

$$\begin{array}{c} K = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix} 5\times 4 \\ \quad\;\; e_1 \quad e_2 \quad e_3 \quad e_4 \end{array}$$

- Examining $T = e_k e_k^T$, we have:

$$T_{uu} = 1, T_{vv} = 1, T_{uv} = -1, T_{vu} = -1$$

$$KK^T = \underset{5\times 4}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}} \underset{4\times 5}{\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}} = \underset{5\times 5}{\begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}} = L$$

GNN from Spectral Perspective

- Property of Laplace matrix
  - $L = KK^T$, Positive semi-definite matrix
  - Eigenvalues ≥ 0
  - Quadratic form of a matrix measures the level of smoothing of a graph

$$f^T L f = ||K^T f||_2^2 = \sum_{(u,v)\in E} (f_u - f_v)^2$$

$$\mathrm{grad}(f) = K^T f = \underbrace{\begin{bmatrix} -2 \\ 2 \\ 1 \\ 1 \end{bmatrix}}_{4\times 1} \begin{array}{l} e_1 : f_1 - f_2 \\ e_2 : f_2 - f_3 \\ e_3 : f_2 - f_4 \\ e_4 : f_4 - f_5 \end{array}$$

GNN from Spectral Perspective

- Define basis from Laplace matrix
  - Quadratic form measures smoothing of a graph

$$f^T L f = ||K^T f||_2^2 = \sum_{(u,v)\in E} (f_u - f_v)^2$$

  - Basis indicate different levels of smoothing like $e^{i2\pi\mu t}$
  - Solve the following problem!

$$u_1 = \underset{||f||=1}{\operatorname{argmin}} f^T L f \qquad u_2 = \underset{||f||=1, f\perp u_1}{\operatorname{argmin}} f^T L f$$

$$u_k = \underset{||f||=1, f\perp span(u_1, u_2 .. u_{k-1})}{\operatorname{argmin}} f^T L f$$

GNN from Spectral Perspective

- Define basis from Laplace matrix
  - $[u_1, u_2, u_3, \ldots, u_\mathrm{n}]$ are the eigenvectors of the matrix L
  - u_k indicates different level of smoothing, $\lambda_k = u_k^T L u_k$
  - Capturing graph structure: $L = \mathrm{D} - \mathrm{A}$
  - Orthogonal
  - Easy calculation
- Fourier transformation on graph

$$\hat{f} = U^T f \qquad f = U \hat{f}$$

GNN from Spectral Perspective

- Normalized graph Laplace matrix
  - Nodes with high degree make too much contribution
  - $\hat{L} = D^{-1}L$ or $\hat{A} = D^{-1}A$
  - $\hat{L} = D^{-1/2}LD^{-1/2}$ or $\hat{A} = D^{-1/2}AD^{-1/2}$

$$f^T\hat{L}f = \sum_{(u,v)\in E}(f_u/d_u - f_v/d_v)^2$$

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2} = \begin{bmatrix} \frac{1}{\sqrt{3}\sqrt{3}} & \frac{1}{\sqrt{3}\sqrt{3}} & \frac{1}{\sqrt{3}\sqrt{4}} & 0 & 0 \\ \frac{1}{\sqrt{3}\sqrt{3}} & \frac{1}{\sqrt{3}\sqrt{3}} & 0 & \frac{1}{\sqrt{3}\sqrt{4}} & 0 \\ \frac{1}{\sqrt{4}\sqrt{3}} & 0 & \frac{1}{\sqrt{4}\sqrt{4}} & \frac{1}{\sqrt{4}\sqrt{4}} & \frac{1}{\sqrt{4}\sqrt{3}} \\ 0 & \frac{1}{\sqrt{4}\sqrt{3}} & \frac{1}{\sqrt{4}\sqrt{4}} & \frac{1}{\sqrt{4}\sqrt{4}} & \frac{1}{\sqrt{3}\sqrt{4}} \\ 0 & 0 & \frac{1}{\sqrt{3}\sqrt{4}} & \frac{1}{\sqrt{3}\sqrt{4}} & \frac{1}{\sqrt{3}\sqrt{3}} \end{bmatrix}$$

$$D = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

GNN from Spectral Perspective

- Convolution on graph

$$f * g = U\left(U^T f \cdot U^T g\right)$$

  - Define kernel on spectral

$$U^T g = [\boldsymbol{\theta}_0, \cdots, \boldsymbol{\theta}_{n-1}]^T$$

$$g_\theta = \operatorname{diag}\left([\boldsymbol{\theta}_0, \cdots, \boldsymbol{\theta}_{n-1}]\right)$$

  - Final formula

$$f * g = U\left(\left(U^T f\right) \cdot \left(U^T g\right)\right)$$

$$f * g = U g_\theta U^T f$$

GNN from Spectral Perspective

- Convolutional network on graph
  - Directly consider θ as a learnable parameter:

$$y = g_\theta(L)x = g_\theta(U\Lambda U^T)x = Ug_\theta(\Lambda)U^T x.$$

$$g_\theta(\Lambda) = \mathrm{diag}(\theta)$$

  - Polynomial parametrization:
    - Utilize the information of eigenvalues
    - Localized filters

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k \Lambda^k,$$

Arxiv'13: Spectral networks and locally connected networks on graphs.

GNN from Spectral Perspective

- Convolutional network on graph
  - Avoiding the calculation over U ($O(n^2)$)
  - Utilizing Chebyshev polynomial of order k

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda}), \qquad \tilde{\Lambda} = 2\Lambda/\lambda_{max} - I_n$$

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x) \text{ with } T_0 = 1 \text{ and } T_1 = x.$$

  - Fast calculation O($K|E|$):

$$y = U \sum_{k=0}^{K} \theta_k T_k(\tilde{\Lambda}) U^T x = \sum_{k=0}^{K} \theta_k T_k(\tilde{L}) x$$

$$\tilde{L} = 2L/\lambda_{max} - I_n \quad \bar{x}_k = T_k(\tilde{L})x \qquad \bar{x}_k = 2\tilde{L}\bar{x}_{k-1} - \bar{x}_{k-2}$$

NIPS'16: Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering

GNN from Spectral Perspective

- GCN: Convolutional +deep neural network
  - not limited to the explicit parameterization
  - Enjoying non-linear deep neural network
  - Let k=2 and $\theta_0 = -\theta_1$

$$g_{\theta'} \star x \approx \theta'_0 x + \theta'_1 \left(L - I_N\right) x = \theta'_0 x - \theta'_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x \,,$$

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

  - Suppose we have a feature matrix:

$$Z = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta \,, \qquad \tilde{A} = A + I_N$$

$$H^{(t)} = \sigma(\hat{\tilde{A}} H^{(t-1)} \Theta)$$

ICLR17': SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS (Citation 2W+)

GNN from Spectral Perspective

- GCN connects spatial and spectral GNN

$$H^{(t)} = \sigma(\hat{\tilde{A}} H^{(t-1)} \Theta)$$

$$\begin{aligned}\mathbf{h}_u^{(k+1)} &= \mathrm{UPDATE}^{(k)}\left(\mathbf{h}_u^{(k)},\ \mathsf{AGGREGATE}^{(k)}\left(\{\mathbf{h}_v^{(k)}, \forall v \in \mathcal{N}(u)\}\right)\right) \\ &= \mathrm{UPDATE}^{(k)}\left(\mathbf{h}_u^{(k)}, \mathbf{m}_{\mathcal{N}(u)}^{(k)}\right),\end{aligned}$$

$$\mathbf{m}_{\mathcal{N}(u)} = \sum_{v \in \mathcal{N}(u)} \frac{\mathbf{h}_v}{\sqrt{|\mathcal{N}(u)||\mathcal{N}(v)|}}$$

$$h_u^{(t)} = \sigma(\Theta^T(\frac{1}{|N(u)|} h_u^{(t-1)} + m_{N(u)})$$

GNN from Spectral Perspective

- Summary of GNN from spectral perspective
  - Convolutional network (Why, How, transfer to graph)
  - Challenge in graph
  - Spectral --- Fourier Transform (basis is important)
  - Basis in graph (what we want? Smoothing!)
  - Define Laplace matrix in graph (estimate smoothing)
  - Basis --- eigenvector of Laplace matrix
  - Different strategies of filtering (efficient!)
  - GCN --- simplified but deep!

Graph Neural Network

....

- Graph neural network
  - Three perspectives
    - Spatial Perspective
    - Spectral Perspective
    - Loss Perspective
  - Applications
  - Promising directions

GNN from Loss Perspective

- Considering the following objective function:

$$\arg\min_{\mathbf{F}} \mathcal{L} = \|\mathbf{F} - \mathbf{S}\|_F^2 + c \cdot tr(\mathbf{F}^\top \mathbf{L}\mathbf{F}).$$

Align with original features

Smoothing

- Optimizing via gradient descent with learning rate b:

$$\mathbf{F}^{(k)} \leftarrow \mathbf{F}^{(k-1)} - b \cdot \left.\frac{\partial \mathcal{L}}{\partial \mathbf{F}}\right|_{\mathbf{F}=\mathbf{F}^{(k-1)}}$$

$$\left.\frac{\partial \mathcal{L}}{\partial \mathbf{F}}\right|_{\mathbf{F}=\mathbf{F}^{(k-1)}} = 2(\mathbf{F}^{(k-1)} - \mathbf{S}) + 2c((I - \widetilde{\mathbf{A}})\mathbf{F}^{(k-1)}$$

$$\mathbf{F}^{(k)} = (1 - 2b - 2bc)\mathbf{F}^{(k-1)} + 2b\mathbf{S} + 2bc\widetilde{\mathbf{A}}\mathbf{F}^{(k-1)}$$

GNN from Loss Perspective

- Considering the following objective function:

$$\arg\min_{\mathbf{F}} \ \mathcal{L} = ||\mathbf{F} - \mathbf{S}||_F^2 + c \cdot tr(\mathbf{F}^\top \mathbf{L}\mathbf{F}).$$

$$\mathbf{F}^{(k)} = (1 - 2b - 2bc)\mathbf{F}^{(k-1)} + 2b\mathbf{S} + 2bc\widetilde{\mathbf{A}}\mathbf{F}^{(k-1)}$$

- Let $b = 1/(2 + 2c)$

$$\mathbf{F}^{(k)} \leftarrow \frac{1}{1+c}\mathbf{S} + \frac{c}{1+c}\tilde{\mathbf{A}}\mathbf{F}^{(k-1)}, k = 1, \ldots K,$$

  - $c \to \boldsymbol{\theta}$ & $\boldsymbol{\theta} \to \infty$,

$$\mathbf{F}^{(k)} \approx \theta\tilde{A}\mathbf{F}^{(k-1)}$$

GNN from Loss Perspective

- Considering the following objective function:

$$\arg\min_{\mathbf{F}} \ \mathcal{L} = \|\mathbf{F} - \mathbf{S}\|_F^2 + c \cdot tr(\mathbf{F}^\top \mathbf{L}\mathbf{F}).$$

$$\mathbf{F}^{(k)} \approx \theta \tilde{A} \mathbf{F}^{(k-1)}$$

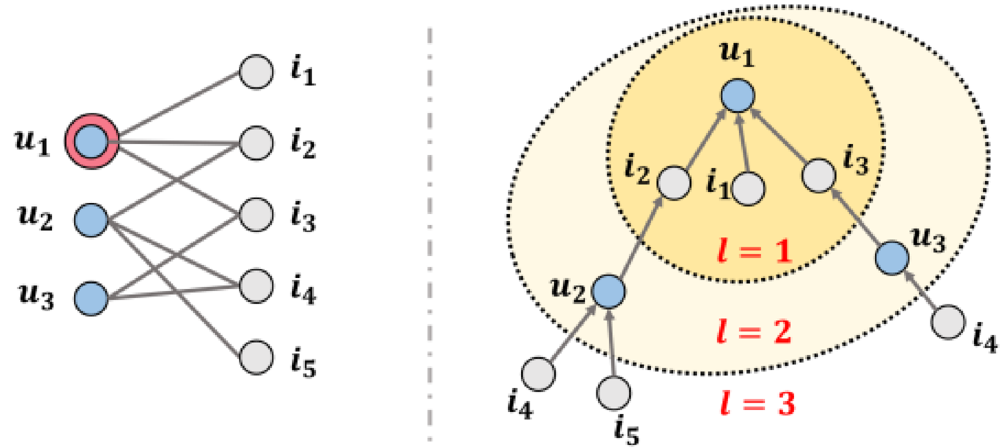
- GCN optimizes smoothing regularizer
- Explanation of over-smoothing with more layers:
  - Decrease the difference between neighbors

Applications

- Keeping motivations in mind:
  - For which data? Relations
  - For which task? Embeddings
- Relations:
  - Any data involves relations
  - Any data involves features? Construct graph!
- Embeddings:
  - Learning representation for each objective

Applications

- Recommendation system



User-Item Interaction Graph

High-order Connectivity for u_1

1	1	0	1
0	1	0	1
0	1	1	1

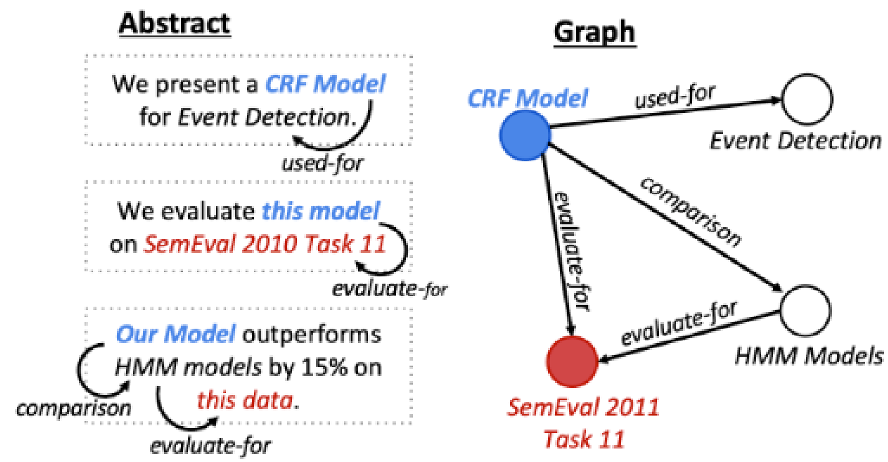
$$\mathbf{E}^{(l)} = \text{LeakyReLU}\Big((\mathcal{L} + \mathbf{I})\mathbf{E}^{(l-1)}\mathbf{W}_1^{(l)} + \mathcal{L}\mathbf{E}^{(l-1)} \odot \mathbf{E}^{(l-1)}\mathbf{W}_2^{(l)}\Big),$$

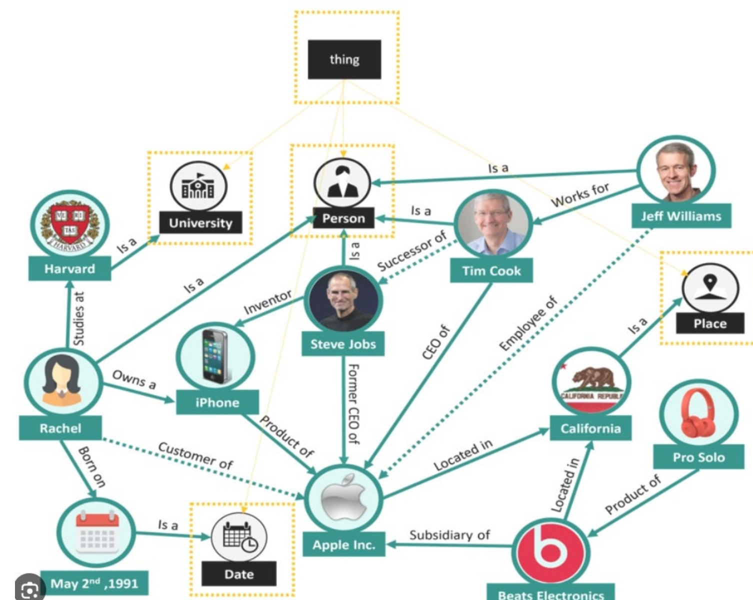
$$\mathcal{L} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \text{ and } \mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^\top & \mathbf{0} \end{bmatrix}$$

SIGIR’19: Neural Graph Collaborative Filtering (citation 2100+)

Applications

- Language
  - Create knowledge graph from text
  - Capturing high-order relations
  - Knowledge graph embeddings

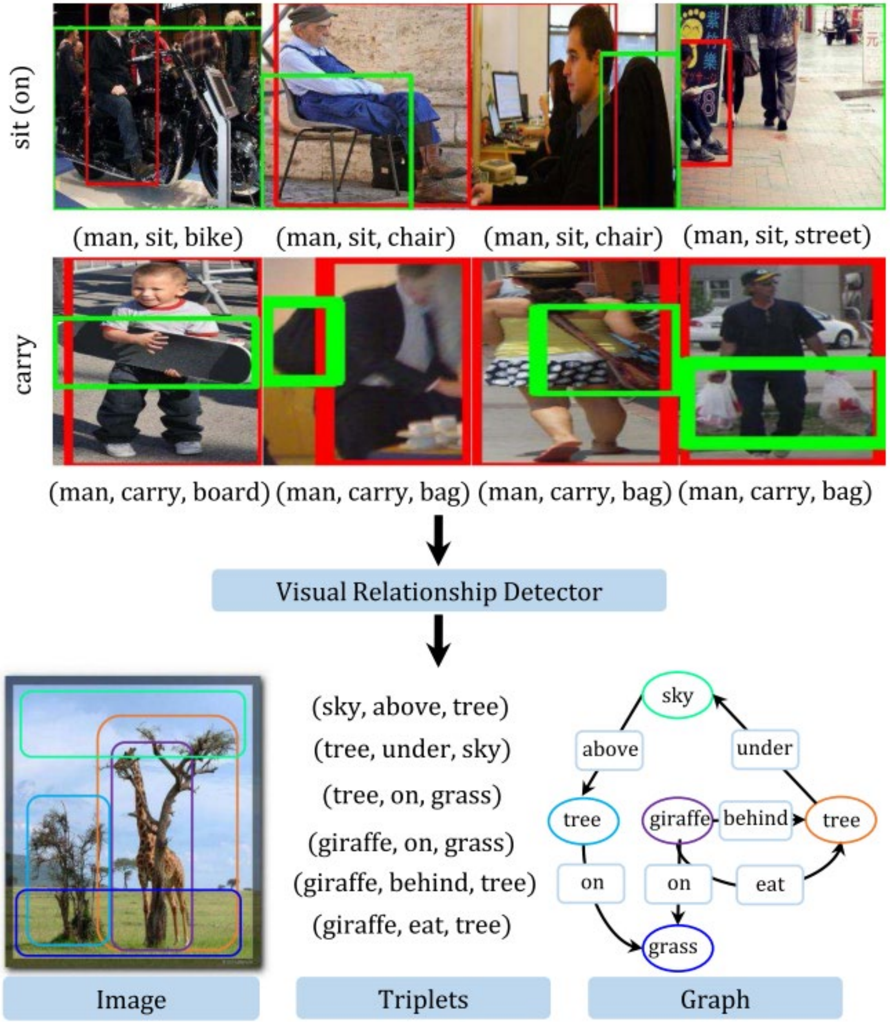




Knowledge graph embedding: A survey of approaches and applications

Applications

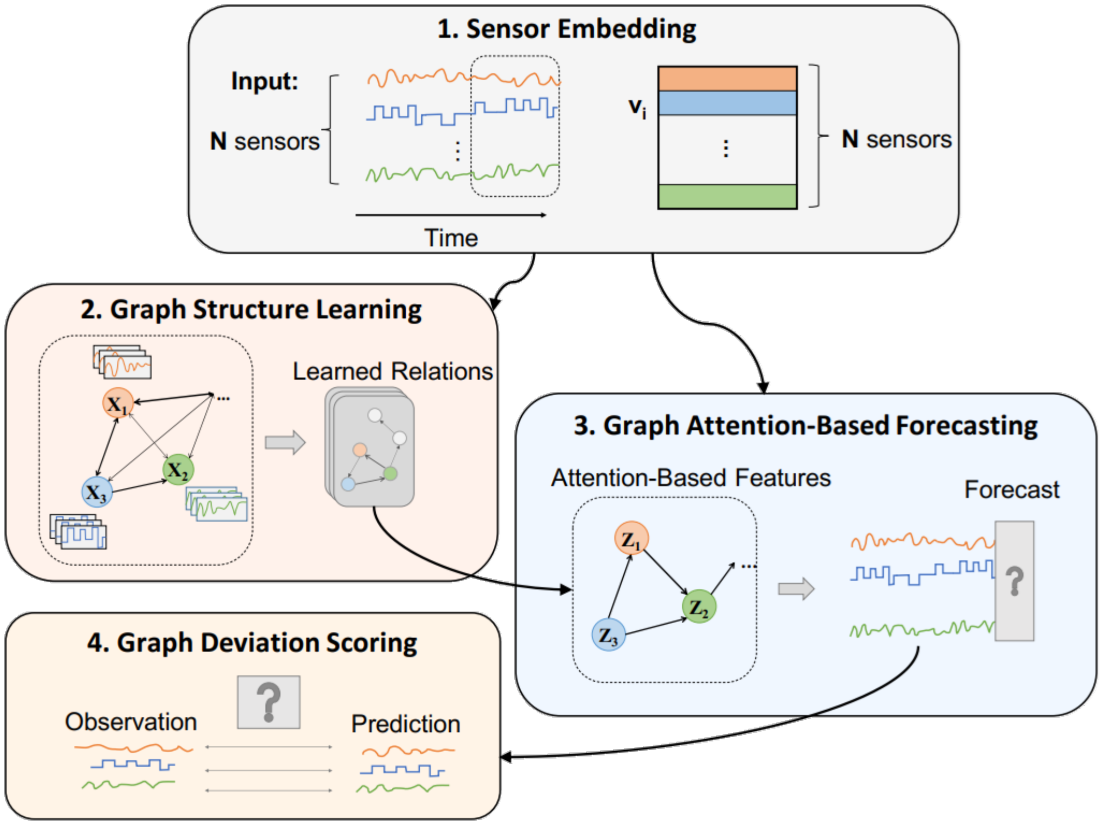
- Computer vision
  - Capturing relations in image
  - Depicts an image via a graph
  - Graph embeddings



CVPR’17: Detecting visual relationships with deep relational networks

Applications

- Anomaly Detection
  - Discovering the relations
  - Construct a graph
  - Learning embeddings
  - Forecasting via estimating deviation scores



Drug discover today 21: A compact review of molecular property prediction with graph neural networks

Graph Neural Network

....

- Graph neural network
  - Three perspectives
    - Spatial Perspective
    - Spectral Perspective
    - Loss Perspective
  - Applications
  - Promising directions

Directions

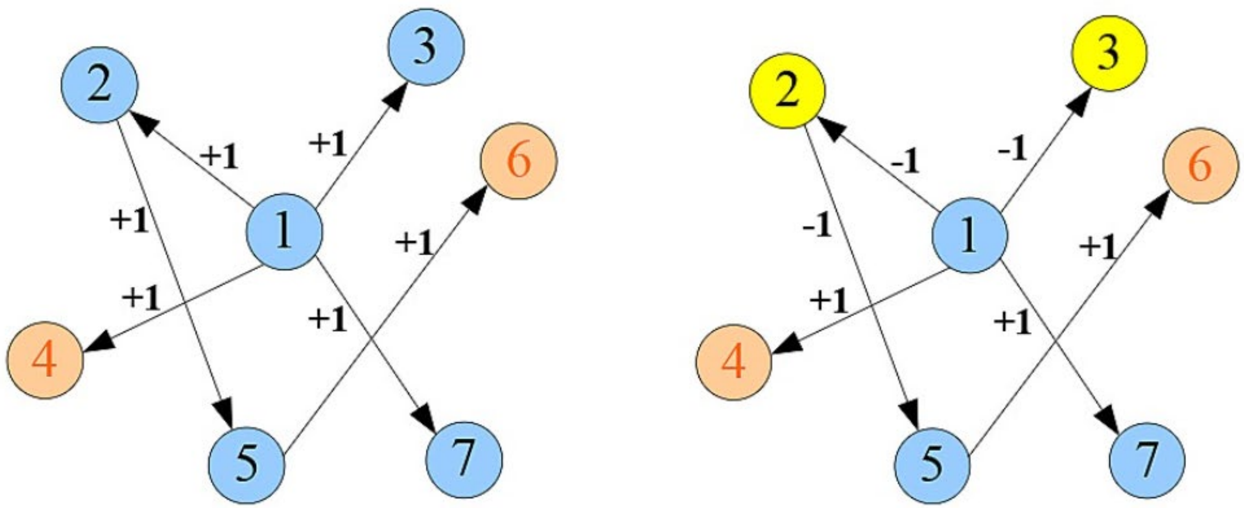
- GNN for complex graph
  - Heterophily graph?

Types	Datasets	# Nodes	# Edges	# Features	# Classes	$\mathcal{H}_{node}$	$\mathcal{H}_{edge}$
WebKB Webpage	Cornell	183	295	1,703	5	0.11	0.30
	Texas	183	309	1,703	5	0.06	0.11
	Wisconsin	251	499	1,703	5	0.16	0.21
Author Co-occurrence	Actor	7,600	33,544	931	5	0.24	0.22
Wikipedia Webpage	Chameleon	2,277	36,101	2,325	5	0.25	0.23
	Squirrel	5,201	217,073	2,089	5	0.22	0.22
	Wiki	1,925,342	303,434,860	600	5	-	0.39
Citation	ArXiv-Year	169,343	1,166,243	128	5	-	0.22
	Snap-patents	2,923,922	13,975,788	269	5	-	0.07
Social Networks	Deezer-Europe	28,281	92,752	31,241	2	0.53	0.53
	Penn94	41,554	1,362,229	5	2	-	0.47
	Twitch-Gamers	168,114	6,797,557	7	2	-	0.55
	Genius	421,961	984,979	12	2	-	0.62
	Pokec	1,632,803	30,622,564	65	2	-	0.45
Webpage Review	YelpChi	45,954	3,846,979	32	2	0.77	0.77

  - **Neighbors can be different !**

Directions

- GNN for complex graph
  - signed graph

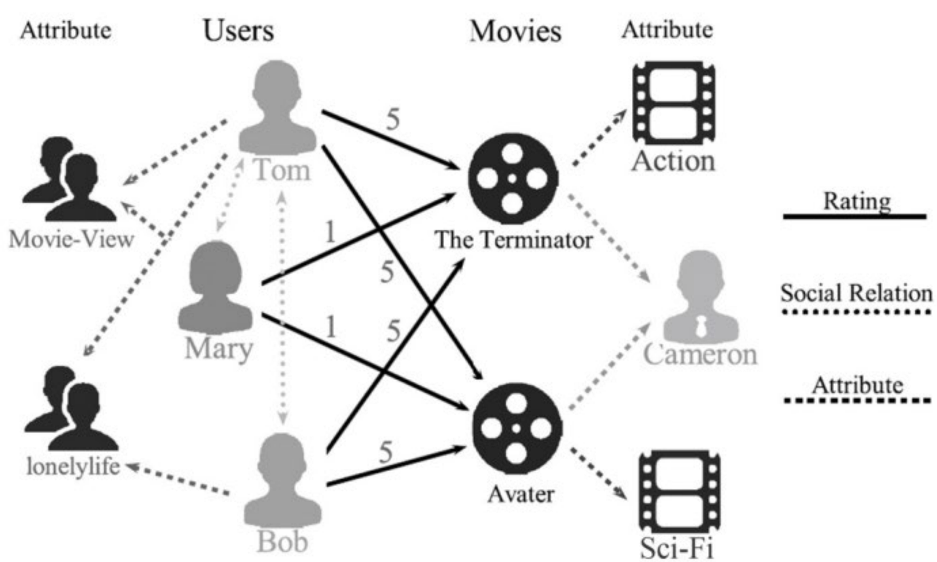


(a) A graph of an unsigned social network (b) A graph of a signed social network

  - Edge=-1 suggests quite dissimilar notes

Directions

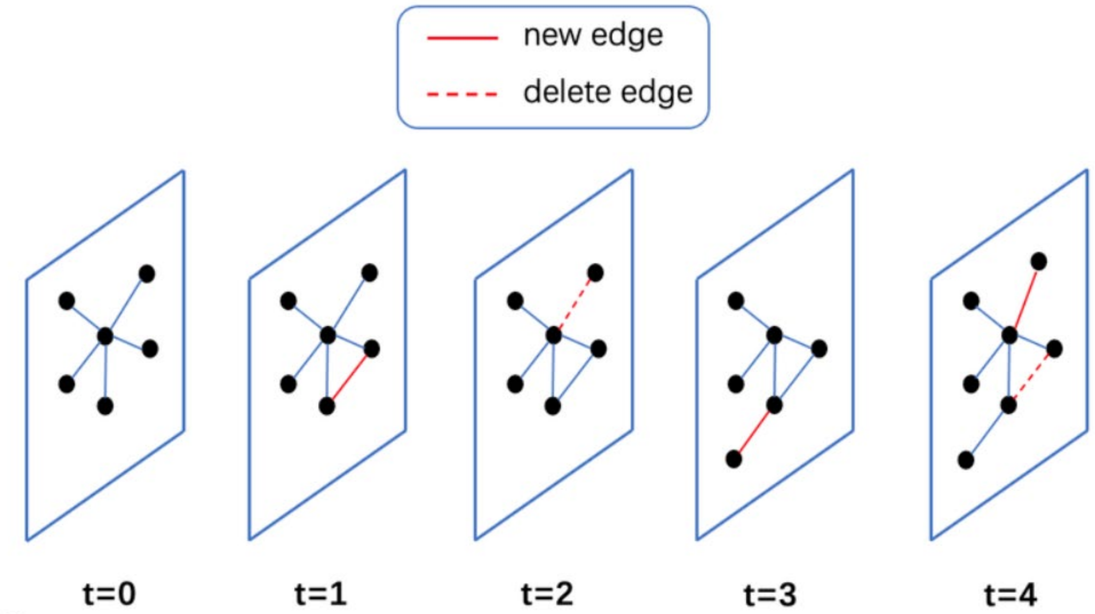
- GNN for complex graph
  - Heterogeneous graph



  - Resigning GNNs for handling different types of nodes and edges

Directions

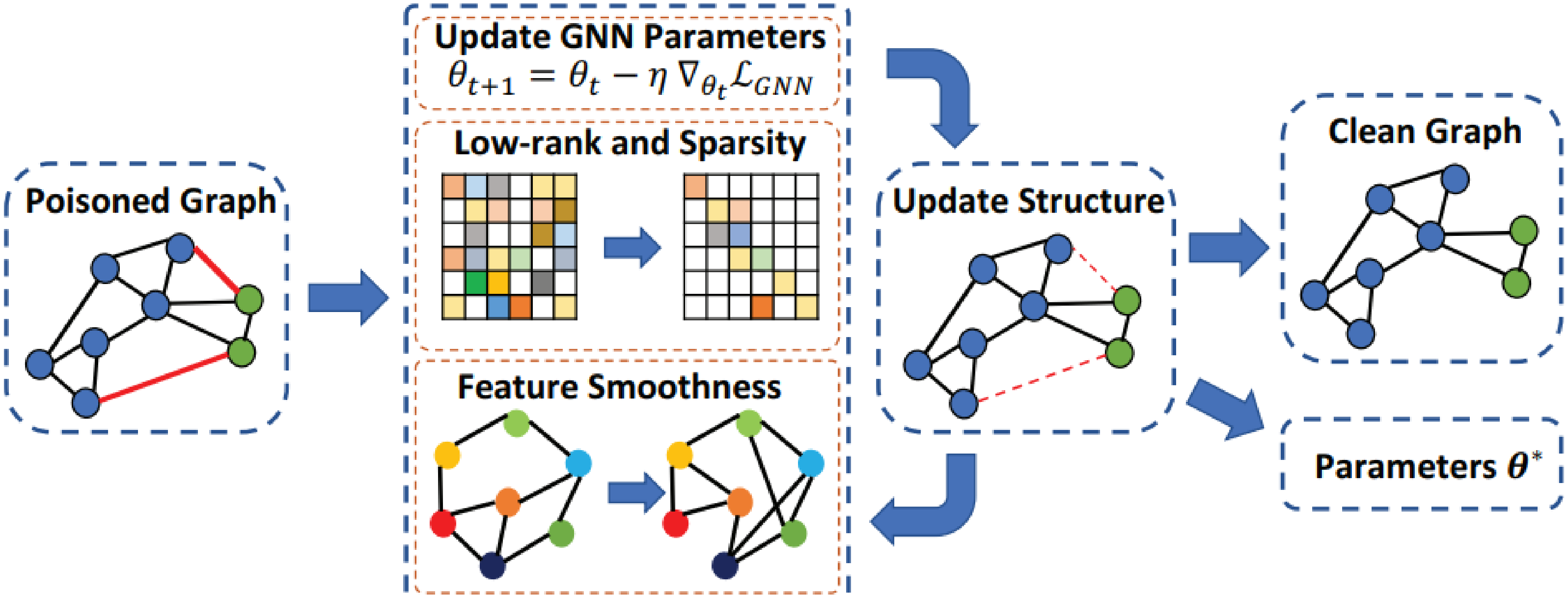
- GNN for complex graph
  - Temporal graph



  - Modeling both temporal patterns and graphic patterns

Directions

- Robust GNN
  - The graph structure is not accurate!

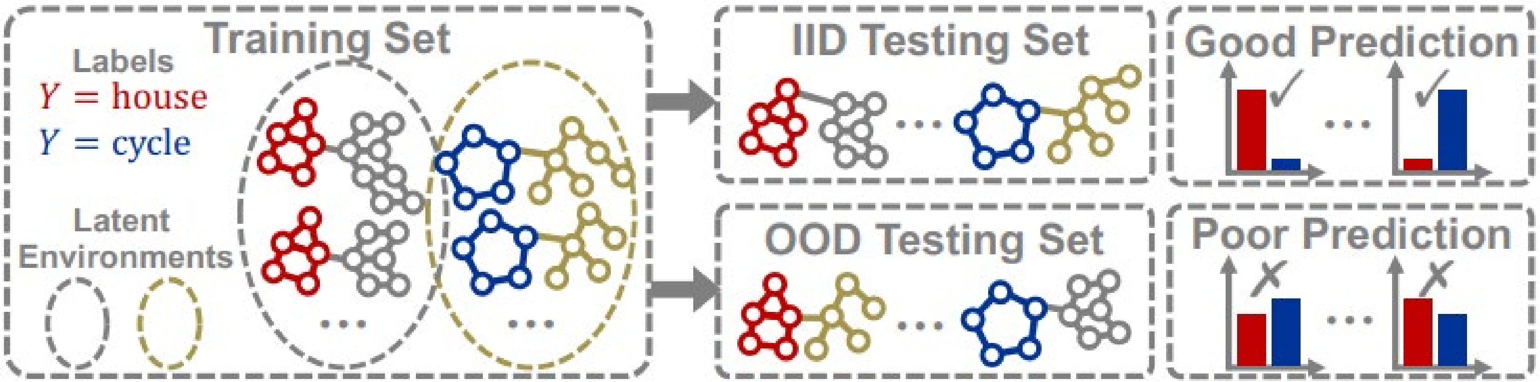


  - Graph structure learning for refining graph.

NIPS23: OpenGSL: A Comprehensive Benchmark for Graph Structure Learning

Directions

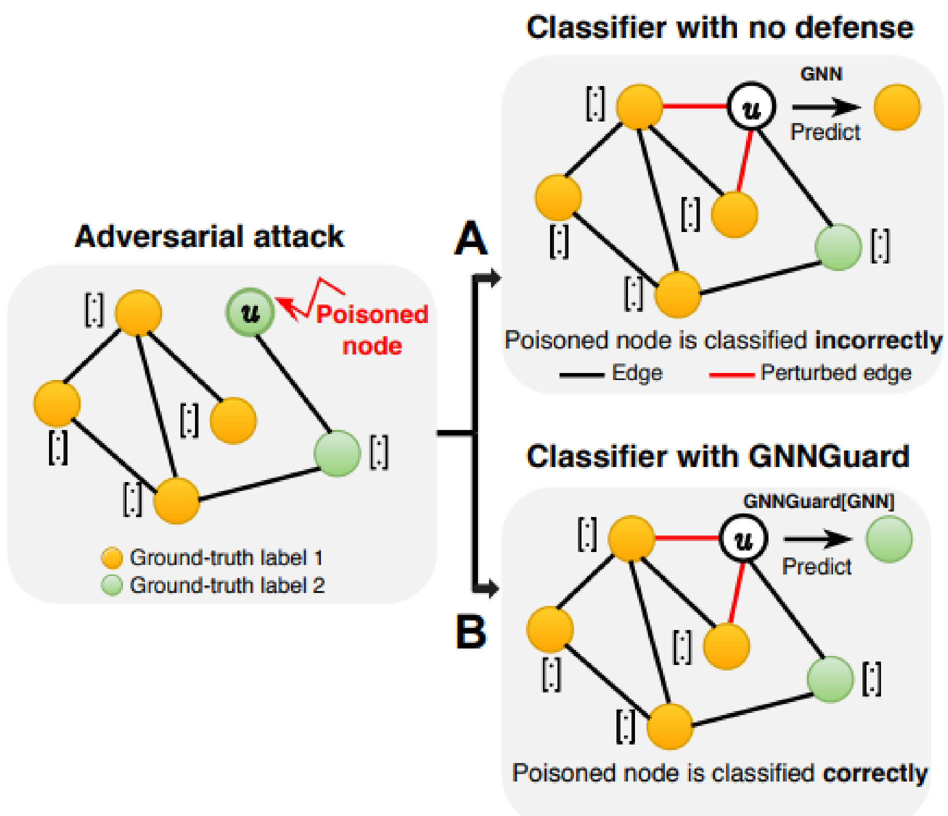
- Robust GNN
  - Distribution shift



  - DRO, invariant learning, causal inference
  - Indentifying important motifs (explanation)

Directions

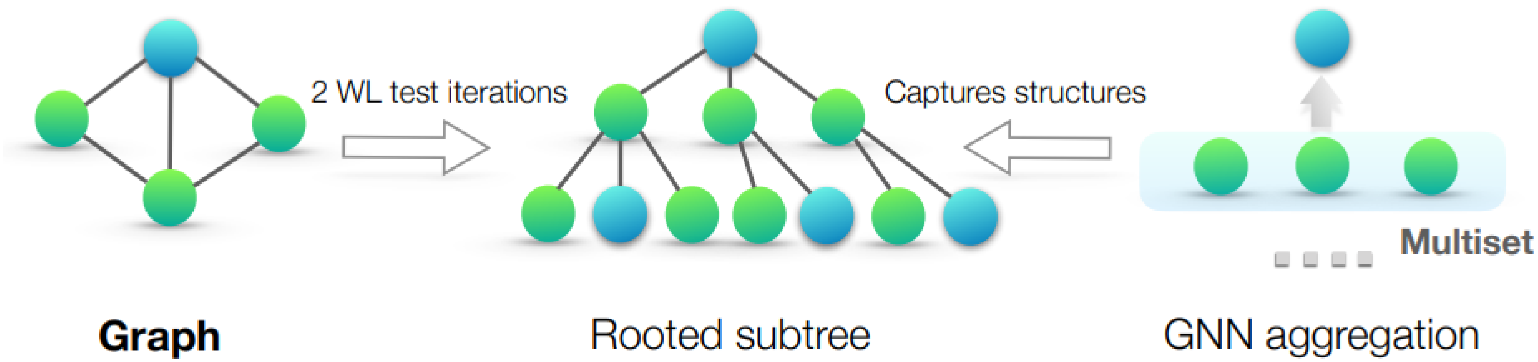
- Robust GNN
  - Attacks



  - Developing different types of attacks and defending accordingly

Directions

- GNN theories
  - Expressive ability



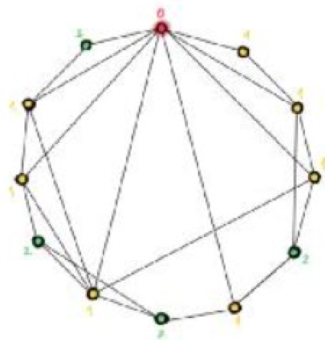
- Connecting GNN with WL test

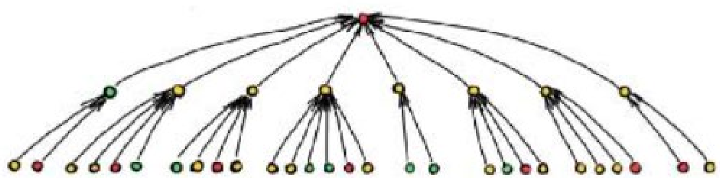
$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left(\left(1+\epsilon^{(k)}\right)\cdot h_v^{(k-1)} + \sum\nolimits_{u\in\mathcal{N}(v)} h_u^{(k-1)}\right)$$

ICLR'19: How powerful are graph neural networks? (Citation 5800+)

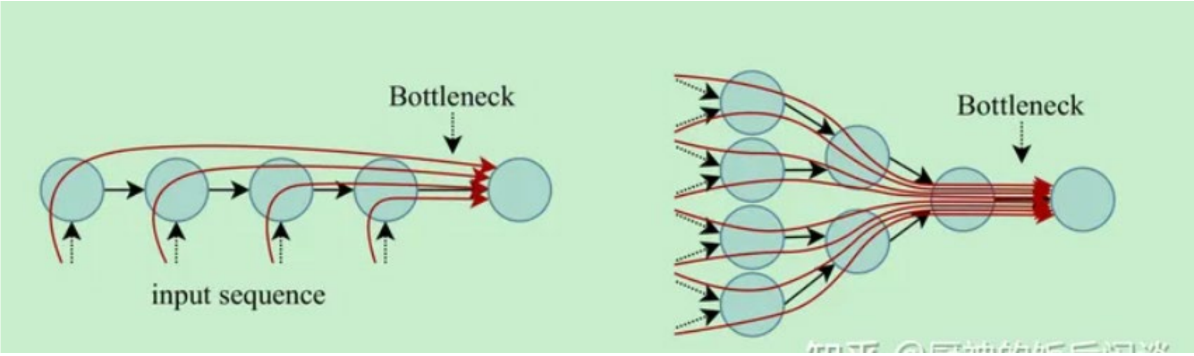
Directions

- GNN theories
  - Over-smoothing
  - Over-squashing





  - bottleneck



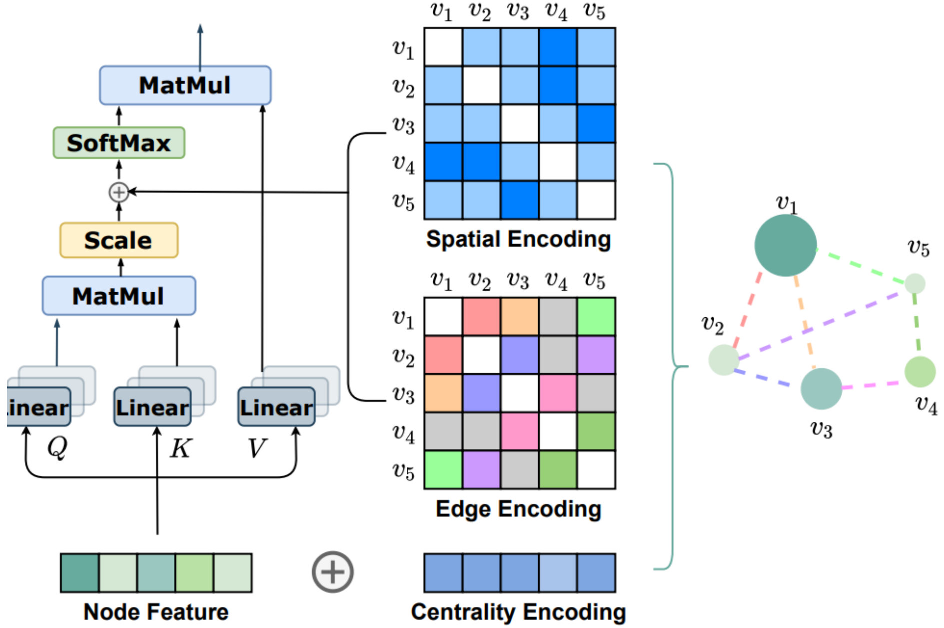
Directions

- Transformer in graph

$$Q = HW_Q, \quad K = HW_K, \quad V = HW_V,$$

$$A = \frac{QK^\top}{\sqrt{d_K}}, \quad \mathrm{Attn}(H) = \mathrm{softmax}(A)V,$$

  - Position encoding
  - Efficiency
  - Combined with GNN



THANK YOU!