# Graph Embeddings

Andrew Liu, Yuhan Xia, You Zhou, Shuai Shao

## Sequence-based

**Graph embedding objective**

$$\min_{\mathbf{Y}} \mathcal{L}(f(\mathbf{A}), g(\mathbf{Y}))$$

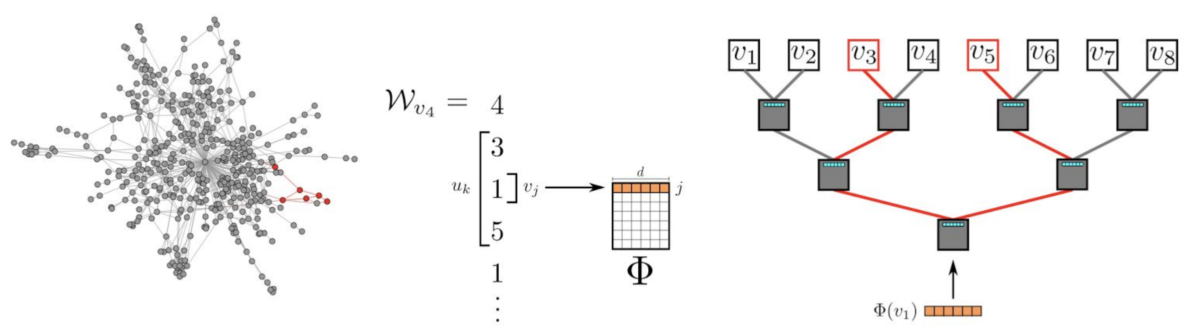
**Random Walk**

- Quality of interest based on Markov-chain:
  - Average first-passage time
  - Average commute time
  - Average first-passage cost
  - Pseudoinverse of the Laplacian matrix
  - Euclidean Commute Time Distance
- Betweeness Centrality

$$b_i = \frac{\sum_{s<t}\sum_j A_{ij} \mid V_i^{(st)} - V_j^{(st)} \mid}{n(n-1)}$$

- Two steps learning
  a. Random walk co-occurrence sampling
  b. Representation learning

**DeepWalk**



**DeepWalk with Graph attention**

- Set based on graph embedding objective

$$g(\mathbf{Y}) = g([\mathbf{L} \mid \mathbf{R}]) = \mathbf{L} \times \mathbf{R}^{\top} \text{ and } f(\mathbf{A}) = \mathbb{E}[\mathbf{D}]$$

- Expectation on the co-occurance matrix

$$\mathbb{E}\left[\mathbf{D}^{\text{DEEPWALK}}; C\right] = \tilde{\mathbf{P}}^{(0)} \sum_{k=1}^{C} \left[1 - \frac{k-1}{C}\right] (\mathcal{T})^k .$$

- Graph attention

$$\mathbb{E}\left[\mathbf{D}^{\text{softmax}[\infty]};\ q_1, q_2, q_3, \ldots\right] = \tilde{\mathbf{P}}^{(0)} \lim_{C\to\infty} \sum_{k=1}^{C} \text{softmax}(q_1, q_2, q_3, \ldots)_k (\mathcal{T})^k ,$$

- Training objectives

$$\min_{\mathbf{L},\mathbf{R},\mathbf{q}} \beta||\mathbf{q}||_2^2 + \left|\left| -\mathbb{E}[\mathbf{D};\mathbf{q}] \circ \log\left(\sigma(\mathbf{L}\times\mathbf{R}^{\top})\right) - \mathbb{1}[\mathbf{A}=0] \circ \log\left(1 - \sigma(\mathbf{L}\times\mathbf{R}^{\top})\right)\right|\right|_1$$

## Hyperbolic embeddings

**Idea** of Hyperbolic Embedding: Complex network graph has potential hyperbolic structure instead of Euclidean Structure.

**Two popular hyperbolic model:**

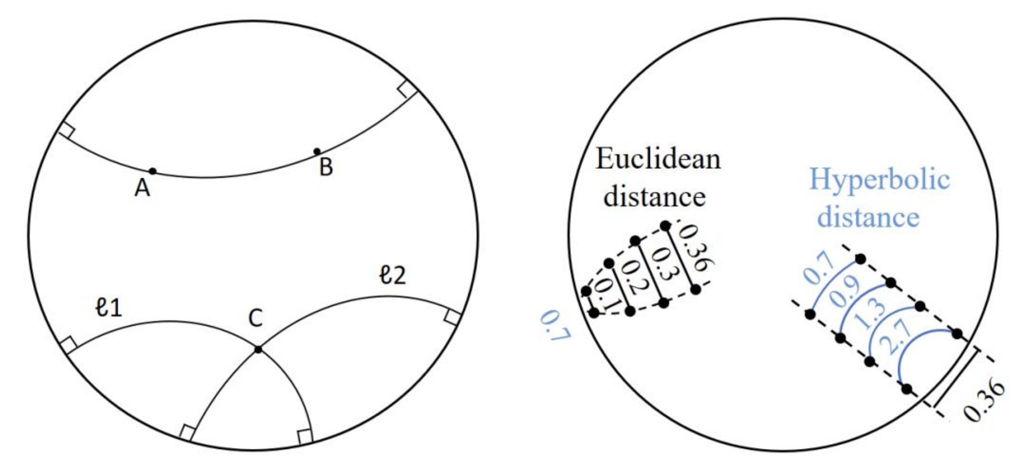
1. *Lorentz Model:*
the upper sheet of a two-sheeted n-dimensional hyperbola

$$< x, y >_{\mathbb{L}} = x^T \mathfrak{g}^L y = -x_0 y_0 + \sum_{i=1}^{n} x_i y_i, x \text{ and } y \in \mathbb{R}^{n+1}$$

$$d(x, y) = \operatorname{arcosh}\left(- < x, y >_{\mathbb{L}}\right).$$

2. *Poincaré Model:*
A unit ball projected from the Lorentz Model

$$d(x, y) = \operatorname{arcosh}\left(1 + 2\frac{\|x-y\|^2}{(1-\|x\|^2)(1-\|y\|^2)}\right).$$




Left: straight lines in Poincare Disk Right: the distance comparison

**Operations in hyperbolic space:**

$$x \oplus y = \frac{(1 + 2\langle x, y\rangle + ||y||^2)x + (1 - ||x||^2)y}{1 + 2\langle x, y\rangle + ||x||^2||y||^2}.$$

Exponential and logarithmic Maps in Lorentz Model:

$$\exp_{\mathbf{x}}(\mathbf{v}) = \cosh(\|\mathbf{v}\|_{\mathcal{L}})\mathbf{x} + \sinh(\|\mathbf{v}\|_{\mathcal{L}})\frac{\mathbf{v}}{\|\mathbf{v}\|_{\mathcal{L}}}$$

$$\log_{\mathbf{x}}(\mathbf{y}) = \frac{\operatorname{arcosh}(-\langle \mathbf{x}, \mathbf{y}\rangle_{\mathcal{L}})}{\sqrt{\langle \mathbf{x}, \mathbf{y}\rangle_{\mathcal{L}}^2 - 1}}(\mathbf{y} + \langle \mathbf{x}, \mathbf{y}\rangle_{\mathcal{L}}\mathbf{x}),$$

where $\|\mathbf{v}\|_{\mathcal{L}} = \sqrt{\langle \mathbf{v}, \mathbf{v}\rangle_{\mathcal{L}}}$.

Exponential and logarithmic Maps in Poincaré Model:

$$\exp_{\mathbf{x}}(\mathbf{v}) = \mathbf{x} \oplus \left(\tanh\left(\frac{\lambda_{\mathbf{x}}\|\mathbf{v}\|}{2}\right)\frac{\mathbf{v}}{\|\mathbf{v}\|}\right)$$

$$\log_{\mathbf{x}}(\mathbf{y}) = \frac{2}{\lambda_{\mathbf{x}}} \operatorname{arctanh}(\|-\mathbf{x} \oplus \mathbf{y}\|)\frac{-\mathbf{x} \oplus \mathbf{y}}{\|-\mathbf{x} \oplus \mathbf{y}\|},$$

**Generalized GNNs based on hyperbolic space:**

$$\mathbf{h}_u^{k+1} = \sigma\left(\sum_{v\in\mathcal{I}(u)} \tilde{\mathbf{A}}_{uv}\mathbf{W}^k\mathbf{h}_v^k\right).$$

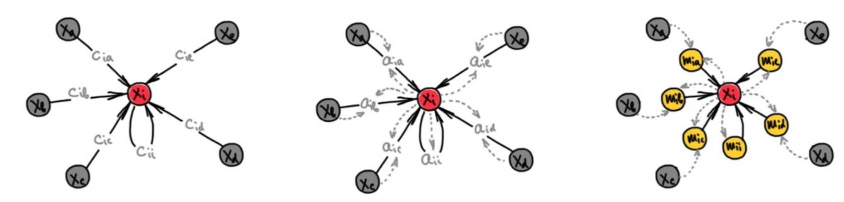
$$\mathbf{h}_u^{k+1} = \sigma\left(\exp_{\mathbf{x}'}\left(\sum_{v\in\mathcal{I}(u)} \tilde{\mathbf{A}}_{uv}\mathbf{W}^k \log_{\mathbf{x}'}(\mathbf{h}_v^k)\right)\right).$$

## Graph neural networks (GNNs)

Idea of GNNs: Permutation-invariant neighborhood aggregations.

Basic GNN categories: convolutional, attentional, general message passing.

$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v\in\mathcal{N}_u} \psi(\mathbf{x}_u, \mathbf{x}_v)\right)$$



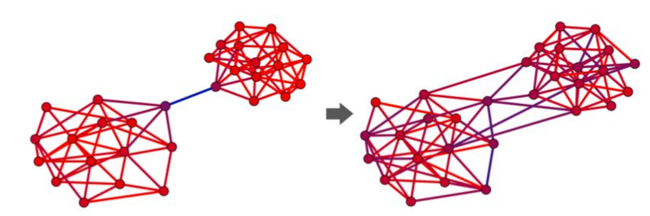
Fundamental limitations of GNNs and respective solutions:

1. Limited expressivity (equivalence to Weisfeiler-Lehman isomorphism test)
   a. Higher-order WL tests
   b. Topological message passing (assign features for simplicial complexes)
2. Oversmoothing
   a. GraphCON dynamical system as a generalized GNN "wrapper".

$$E(X) = \frac{1}{|V|}\sum_{i\in V}\sum_{j\in N_i}\left\|X_i - X_j\right\|^2.$$

$$X'' = \sigma\left(F_\theta(X, t)\right) - \gamma X - \alpha X'$$

3. Oversquashing/bottlenecks
   a. Curvature-based graph rewiring



$$\text{Ric}(i, j) := \frac{2}{d_i} + \frac{2}{d_j} - 2 + 2\frac{|\sharp_\Delta(i,j)|}{\max\{d_i, d_j\}} + \frac{|\sharp_\Delta(i,j)|}{\min\{d_i, d_j\}} + \frac{(\gamma_{max})^{-1}}{\max\{d_i, d_j\}}(|\sharp_\square^i| + |\sharp_\square^j|),$$

   b. Graph neural diffusion (GRAND)

Another GNN generalization that solves oversmoothing, bottlenecks,and stability problems.

$$\frac{\partial x}{\partial t} = \nabla \cdot (g(u, x(u,t), t)\nabla x(u,t))$$

$$\frac{\partial x}{\partial t} = (A(x(t)) - I)x(t) \equiv \bar{A}(x(t))x(t)$$

Choice of discretization schemes leads to different GRAND architectures.