Attributed network embedding

- Motivations & challenges
- Mining attributed networks with shallow embedding
- **Mining attributed networks with deep embedding**
  - Objective function based deep embedding
  - Graph neural networks
- Human-centric network analysis

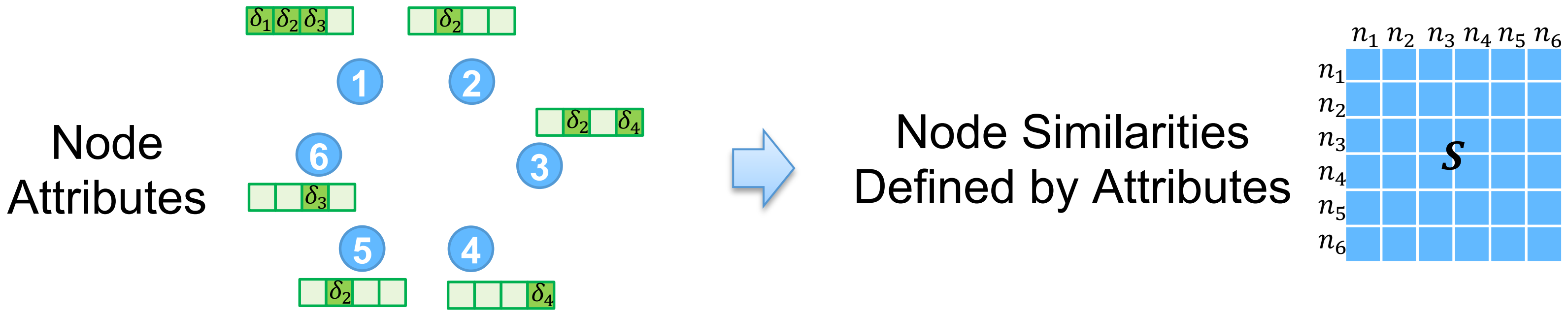
Objective function based deep embedding

- Objective function of DeepWalk:

$$\mathcal{J}_{\text{DeepWalk}} = -\log(\sigma(\mathbf{h}_u^\top \mathbf{h}_v)) - Q \cdot \mathbb{E}_{v_n \sim P_n(v)} \log(\sigma(-\mathbf{h}_u^\top \mathbf{h}_{v_n}))$$

- v is a node that co-occurs near u on fixed-length random walks
- σ is the sigmoid function. Q is the number of negative samples
- $P_n(v)$ is a negative sampling distribution, based on the node frequencies in the entire node sequences
- It trains a unique embedding representation for each node via a representation look-up table
- How to incorporate node attributes in deep architectures?

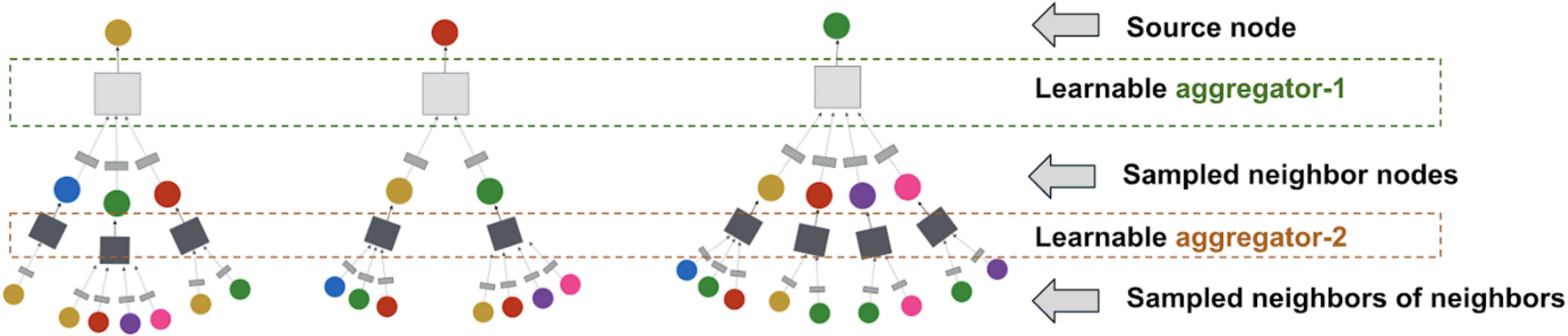
Property preserving network embedding



- Compute the node similarity matrix $\boldsymbol{S}$ defined by node attributes
- Objective function: $\mathcal{J} = \mathcal{J}_{\text{DeepWalk}} + \sum_{i \in \text{pos}(v) \cup \text{neg}(v)} d(v, i)$
- $\mathbf{S}_{vi}$ is the attribute similarity between u and v
- $d(v, i) = \sqrt{(\mathbf{h}_v - \mathbf{h}_i)^\top (\mathbf{h}_v - \mathbf{h}_i)}$ measures distance in embedding space
- $\text{pos}(v)$ and $\text{neg}(v)$ are sets of top-k similar and dissimilar nodes according to $\boldsymbol{S}$

Li et al., "PPNE: Property Preserving Network Embedding", DASFAA, 2019.

Graph neural networks

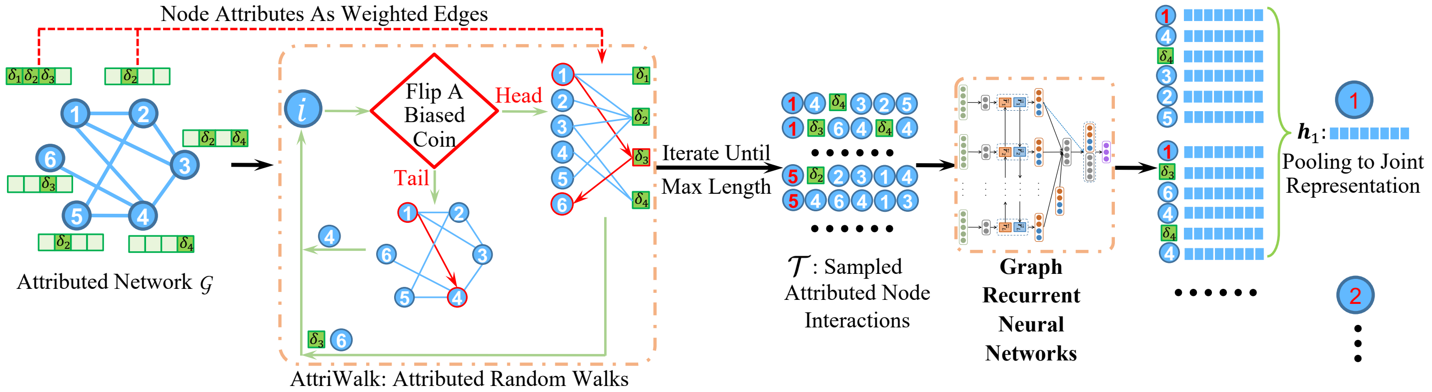


- Key ideas of graph convolutional networks and GraphSage:
  - Use node attributes or random vectors as initial latent representations
  - Each node's representation is learned via averaging its neighbors' representations in previous layer

- It could be considered as a first-order approximation of spectral graph convolutions

Hamilton et al., "Inductive Representation Learning on Large Graphs", NIPS, 2017.

Graph recurrent networks with attributed walks

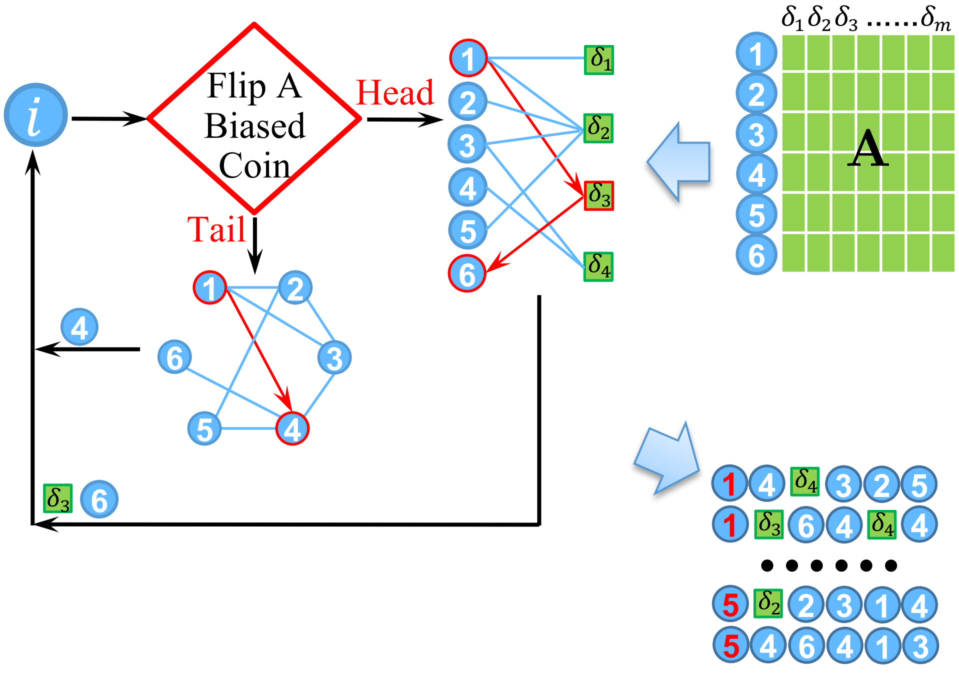


GraphRNA [Huang et al. KDD, 2019]

- A unified walking mechanism is proposed to jointly sample networks and node attributes
- Graph recurrent network (GRN) could preserve node order information
- Nodes are allowed to interact in GRN via the same way as they interact in the original attributed network

A joint walking mechanism - AttriWalk

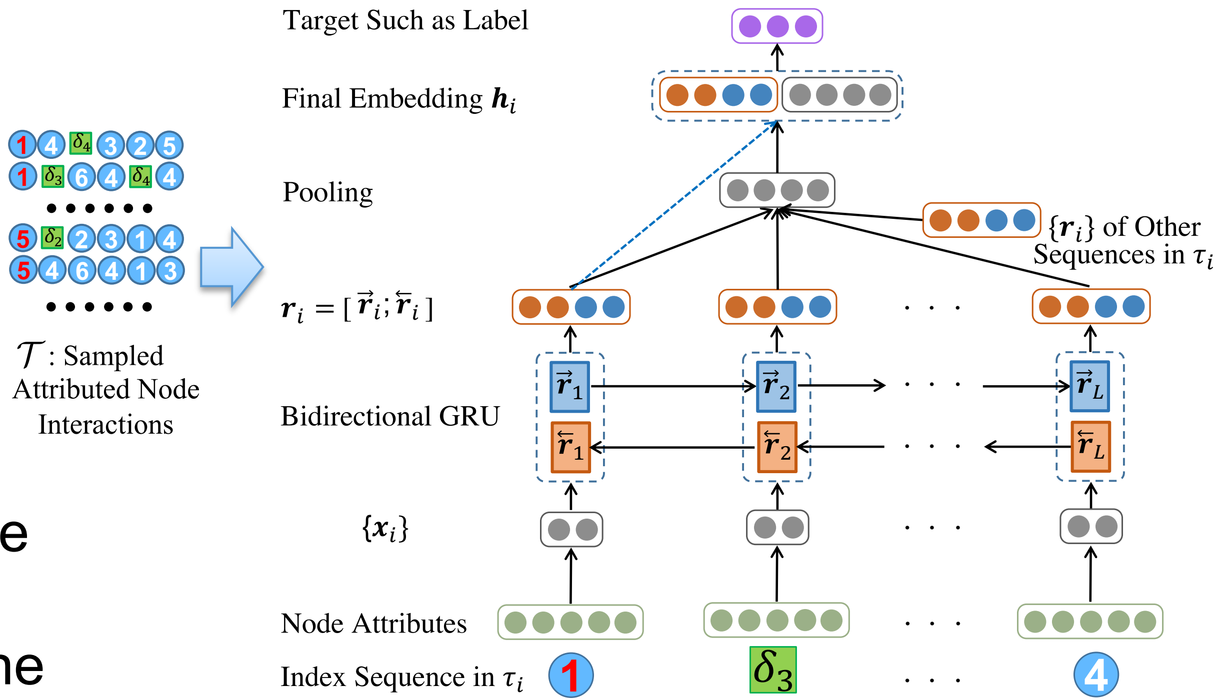
- Construct a bipartite network based on $\mathbf{A}$
- Flip a biased coin in each step
- If head, walk two steps on the bipartite network
  - Jump to an attribute category δ_k
  - From δ_k, jump to a node j
- If tail, walk one step on the original network $\mathbf{G}$
- Walks on $\mathbf{G}$ inherit properties of traditional random walks; walks on $\mathbf{A}$ increase the diversity and flexibility



$\mathcal{T}$: Sampled Attributed Node Interactions

Graph recurrent neural networks - GRN

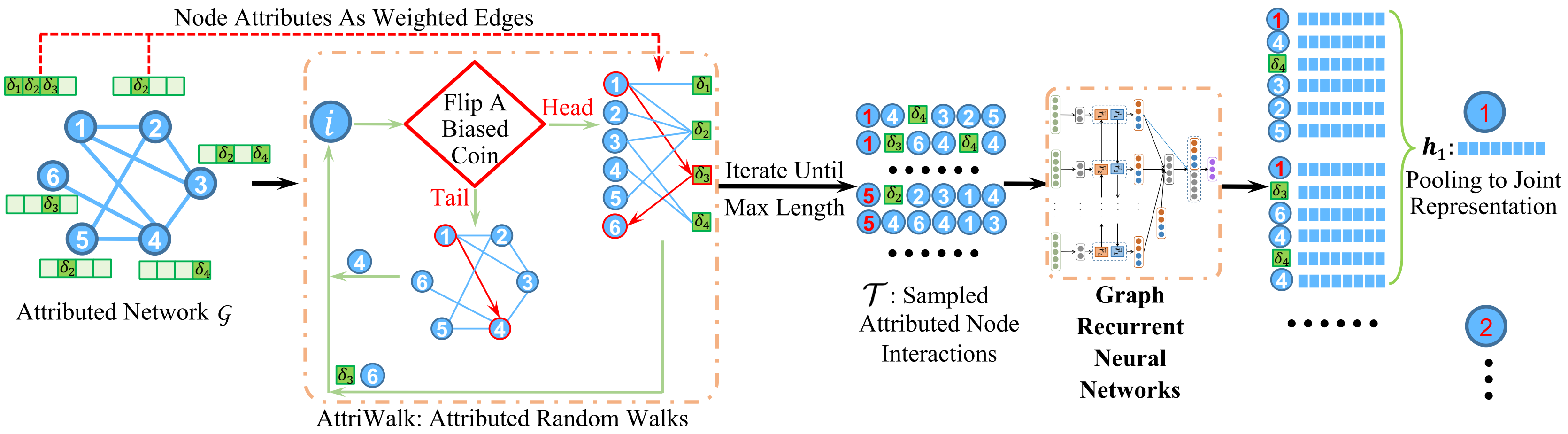
- Hidden state sequences in RNN naturally accord with sampled node interactions



- Pooling layers combine indices within each sequence, and combine all sequences of each node

- It concatenates the first embedding representation for self loop

Task-specific objective function & multiple sources



- GraphRNA could be trained with an unsupervised, supervised, or task-specific objective functions, e.g.,

$$\mathcal{L} = -\sum_{i \in \mathcal{V}} \mathbf{y}_i^\top \log(\text{softmax}(\sigma(\mathbf{h}_i \mathbf{W}_h + \mathbf{b}_h)))$$

- Graph neural networks could be an embedding model or an end-to-end model for different tasks

Mining attributed networks with deep embedding

- Focuses:
  Deep architectures for networks & joint learning
- Methods:
  Objective function based deep embedding
  Graph neural networks
- Architectures:
  Graph convolutional networks
  Graph recurrent networks