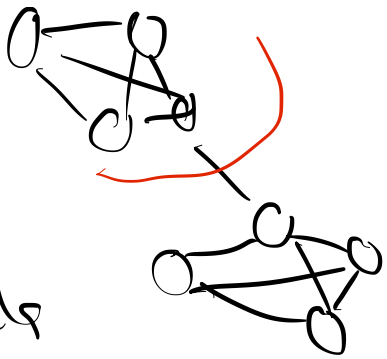
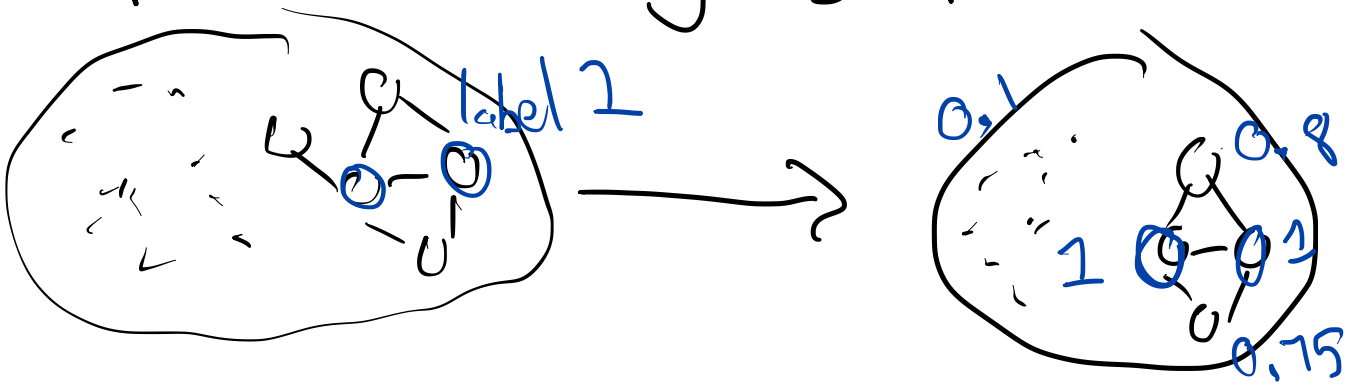
April 9, 2020



"unsupervised" learning: graph clustering

"semi-supervised" learning: graph + some labels

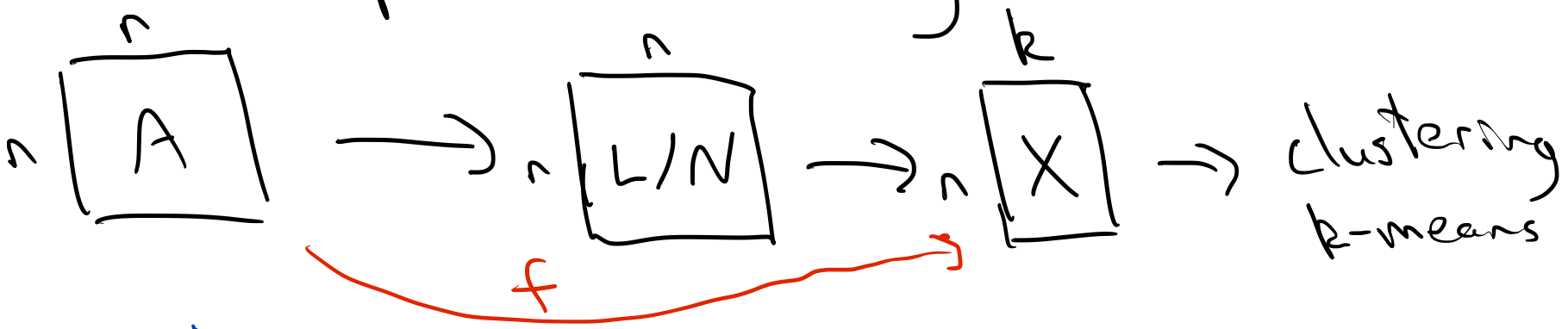


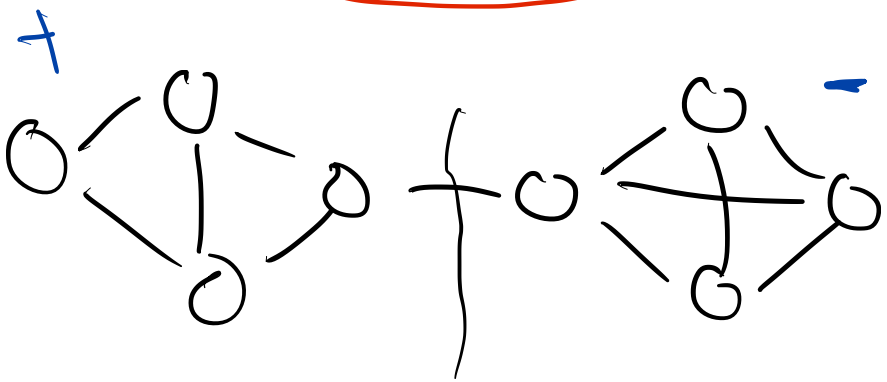
Today: "representation" learning on graphs

$$n \left[\boxed{A}\right] \xrightarrow{f} n \left[\boxed{X}\right] \in \mathbb{R}^{n \times k}$$

- generative models
- data interpretation
- "downstream" ML tasks

Example: spectral clustering





Case 1: Latent space models (Hoff et al. 02)

$x_1, \ldots, x_n$ latent "positions of n nodes $x_i \in \mathbb{R}^k$

Idea: nearby nodes are likely to connect

$$p_{ij} = P_r(i \to j) = \frac{1}{1 + \exp(\|x_i - x_j\|_2^2 - \alpha)}$$

$$\text{logit}(p_{ij}) = \log\left(\frac{p_{ij}}{1-p_{ij}}\right) = \alpha - \|x_i - x_j\|_2^2$$

Social networks perspectives:

① $i \to j \Rightarrow x_i, x_j$ close $\Rightarrow j \to i$ likely (reciprocity)

② $i \to j, i \to k \Rightarrow x_j, x_k$ close to $x_i \Rightarrow x_j, x_k$ close

$\Rightarrow j \leftrightarrow k$ likely $\Rightarrow$ triangles

Max likelihood: $\max_{\{x_i\}, \alpha} \log(P_r(A \mid \{x_i\}, \alpha))$

$$= \sum_{(i,j) \in E} \log(P_r((i,j) \mid x_i, x_j, \alpha)) + \sum_{(i,j) \notin E} \log(1 - P_r((i,j) \mid x_i, x_j, \alpha))$$

Incorporate node/edge data

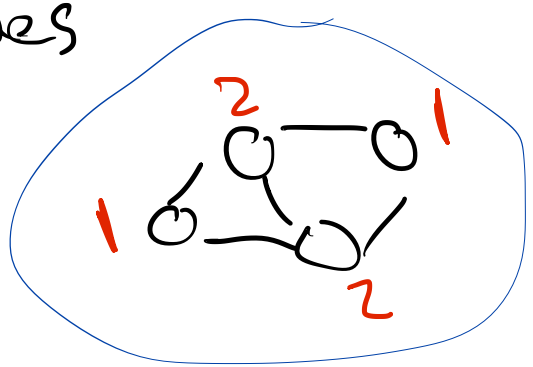
$$\text{logit}(p_{ij}) = \log\left(\frac{p_{ij}}{1-p_{ij}}\right) = \alpha - \|x_i - x_j\|_2^2 + \beta z_{ij} + \gamma_1 z_i + \gamma_2 z_j$$

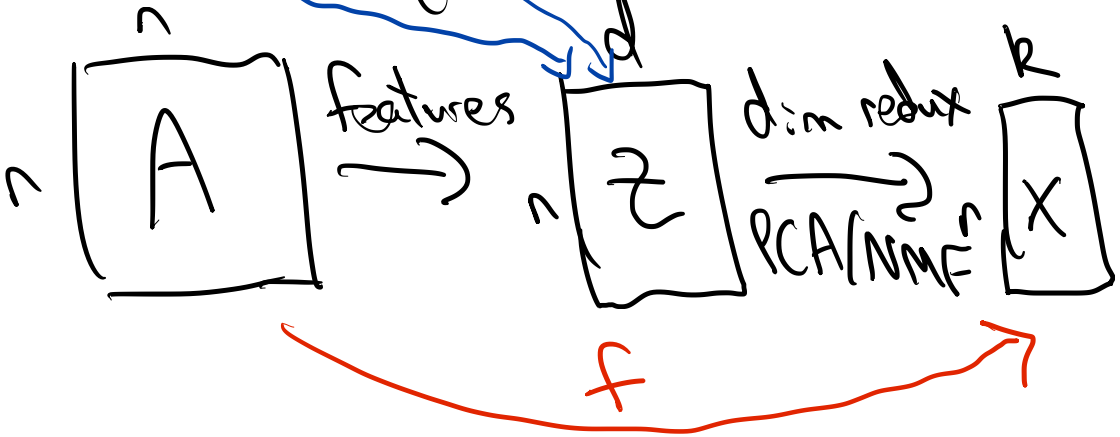
Case 2: RolX (Henderson et al. 2012)

Idea: rather than worry about edges directly, ask what graph structure says about nodes

① Collect features for nodes

- degree
- number triangles
- PageRank
- squared degree





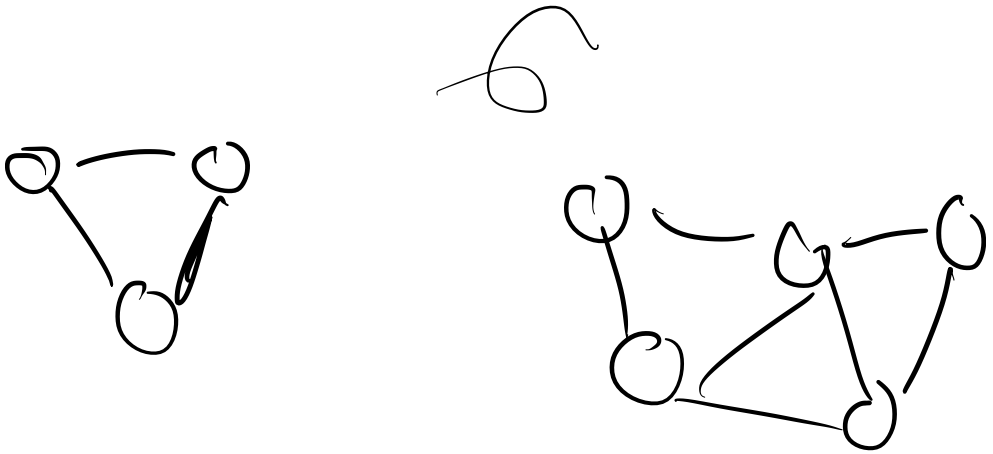
② cluster in "feature space"

k-means with X

"roles" vs. "clusters"

Also check out: ERGMs

$$p_\beta(G) \propto \exp(-\beta^T x)$$



Case 3: node2vec (Grover and Leskovec 2016)

Goal: learn general-purpose representations that are hopefully good for various "downstream" ML tasks
- → clustering
- → link prediction

model: $\max_{\{x_i\}} \prod_i \Pr(N_S(i) \mid x_i)$

representation

sampled "neighborhood"

Assumptions: ① $\Pr(N_S(i) \mid x_i) = \prod_{j \in N_S(i)} \Pr(j \mid x_i)$

② $\Pr(j \mid x_i) = \exp(x_j^T x_i) / \sum_k \exp(x_k^T x_i)$

$$-\frac{1}{2}\|x_i - x_j\|_2^2 = -\frac{1}{2}\|x_i\|_2^2 - \frac{1}{2}\|x_j\|_2^2 + x_j^T x_i$$

$$\simeq -1 + x_j^T x_i$$

$$\max_{\{x_i\}} \sum_i \sum_{j \in N_s(i)} x_j^T x_i - \log Y_i$$

$$Y_i = \sum_k \exp(x_k^T x_i)$$

Idea: negative sampling

Replace $\sum_{k \in V} \exp(x_k^T x_i)$ with $\sum_{k \in W} \exp(x_k^T x_i)$

W sample of nodes, fixed size $|W| = 25$

$$P(j \mid x_i) \approx \frac{\exp(x_j^T x_i)}{\sum_{k \in W} \exp(x_k^T x_i)}$$

How do I sample $N_S(i)$?

Two main strategies:

1. BFS
2. DFS