Relational/Graph Learning

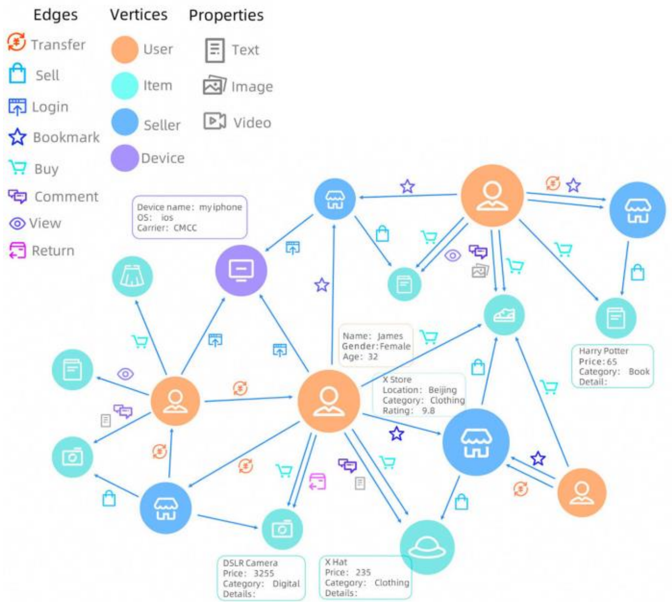
Teacher: Zheng Wang

Social Network Analysis (NIS8023)

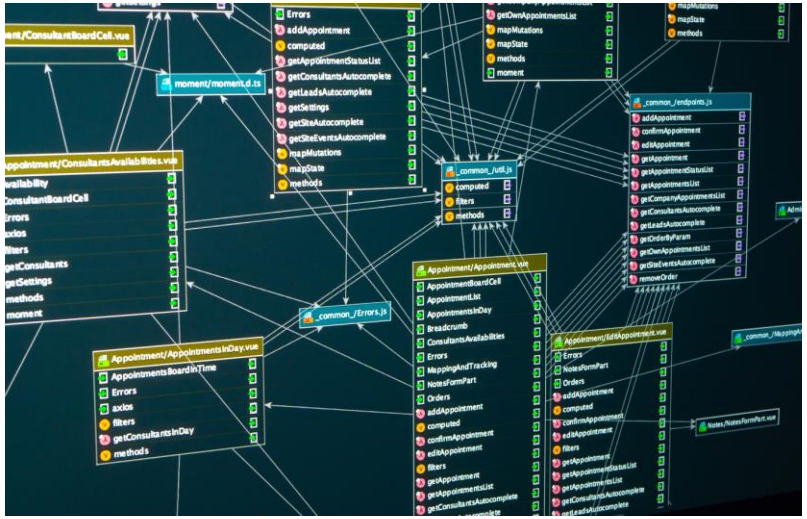
Shanghai Jiao Tong University

Relational/Graph Data are Everywhere

Social networks



E-commerce

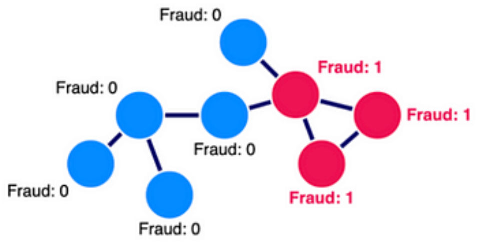


Relational databases

…

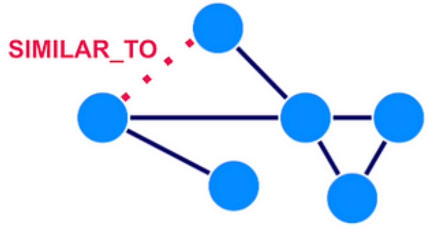
The Aim of Graph Learning

- Graph learning, machine learning on graphs, aims to learn complex relationships among nodes and the topological structure of graphs.



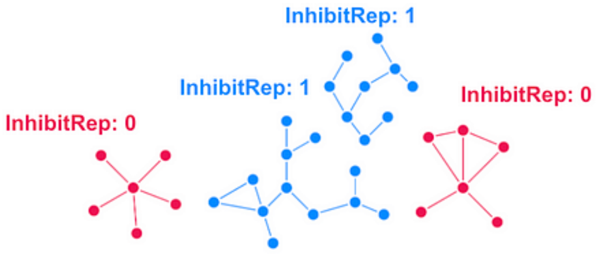
Node Property Prediction

Predict a discrete or continuous node property, called **node classification** and **node regression** respectively.



Link Prediction

Predict if a relationship should exist between two nodes. Often a binary classification task, but can sometimes include more link types or continuous properties.



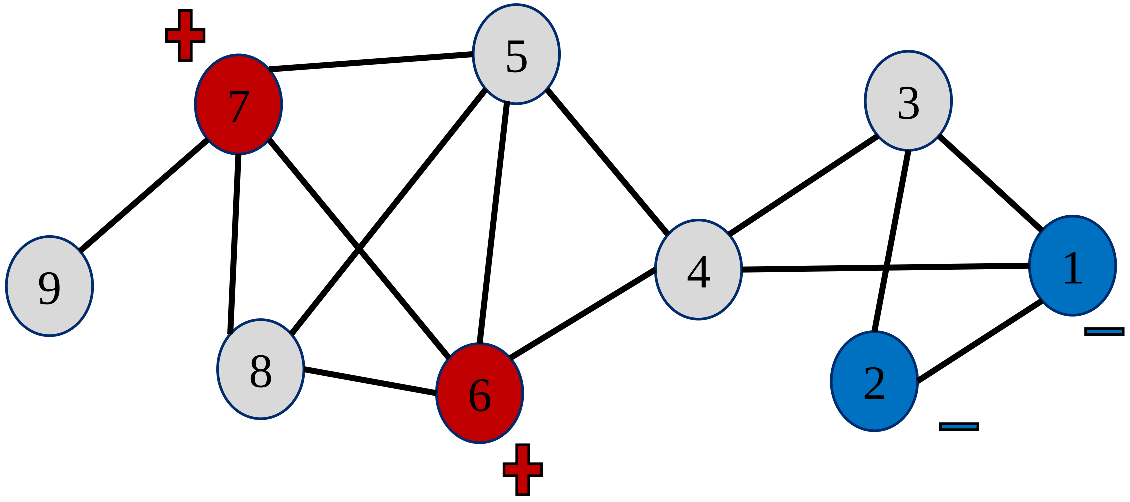
Graph Property Prediction

Predict a discrete or continuous property of a graph or subgraph.

https://medium.com/towards-data-science/graph-machine-learning-an-overview-c996e53fab90

A Graph Node Classification Example

- **Objective**: Exploit the structural information within a graph to enhance the prediction of node attributes or labels.

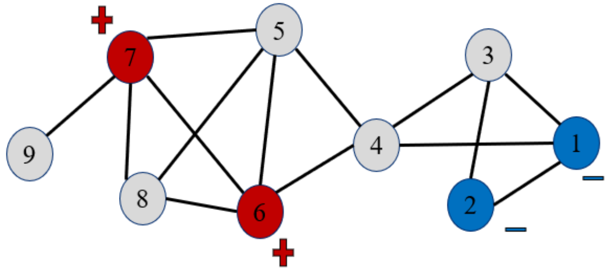


Given the graph structure with some labeled nodes (i.e., 1, 2, 6, 7 are labeled), predict the labels for the remaining nodes.

Outline

- **Shallow Methods**
  - Weighted-vote relational neighbor
  - Label Propagation
- Deep Learning Methods

Recap: Traditional Non-Graph Learning Methods

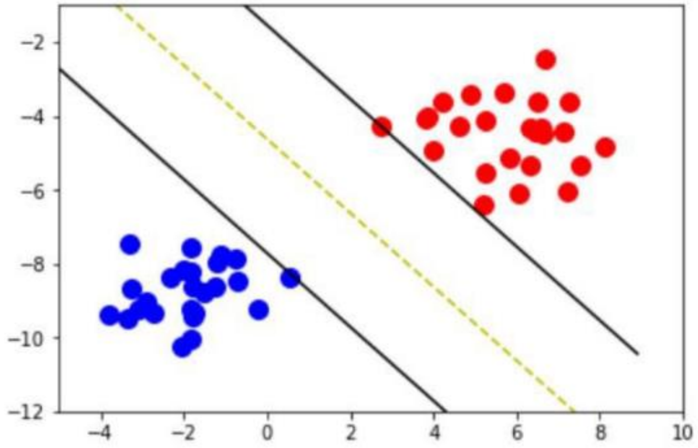


Node features (by edge)

$$u_1 = [0, 1, 1, 1, 0, 0, 0, 0, 0]$$
$$u_2 = [1, 0, 1, 0, 0, 0, 0, 0, 0]$$
$$\vdots$$

Node labels

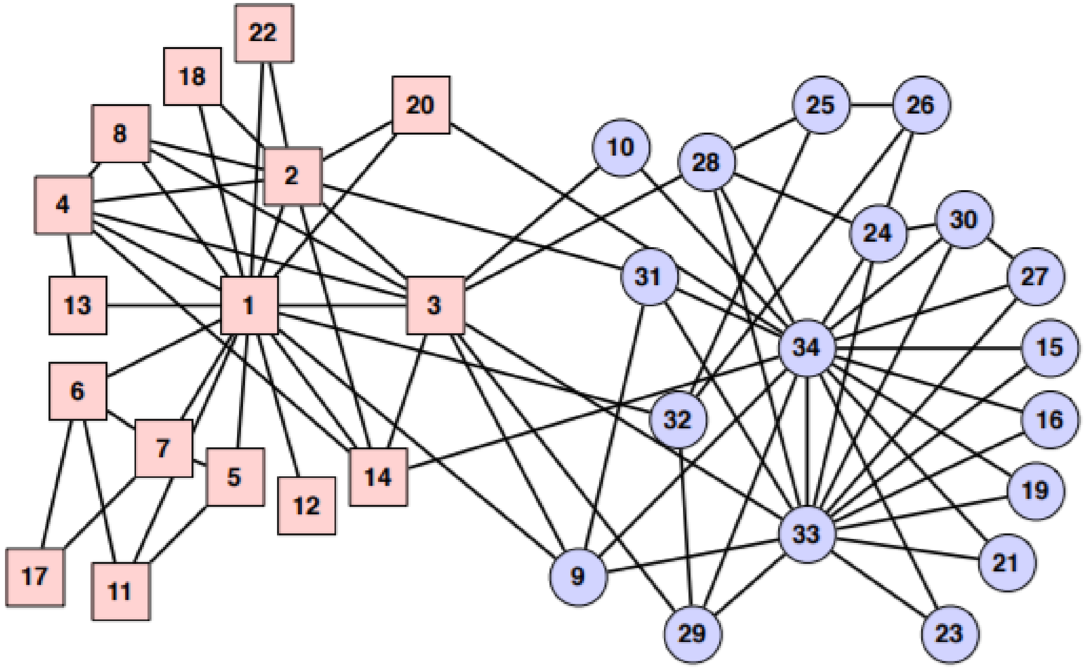
$$y_1 = 0$$
$$y_2 = 0$$
$$y_6 = 1$$
$$y_7 = 1$$



Common classifiers: (like linear classifiers and SVMs)

How To Use Graph Information Effectively?

- Observation: Linked nodes tend to have similar labels



The karate club with ground truth

The Key Idea Is "Collective Classification"

- Definition: Perform collaborative classification of unlabeled nodes in a partially labeled network using techniques such as label propagation or approximate inference [1]. Two typical methods:
  - Weighted-vote relational neighbor
  - Label propagation

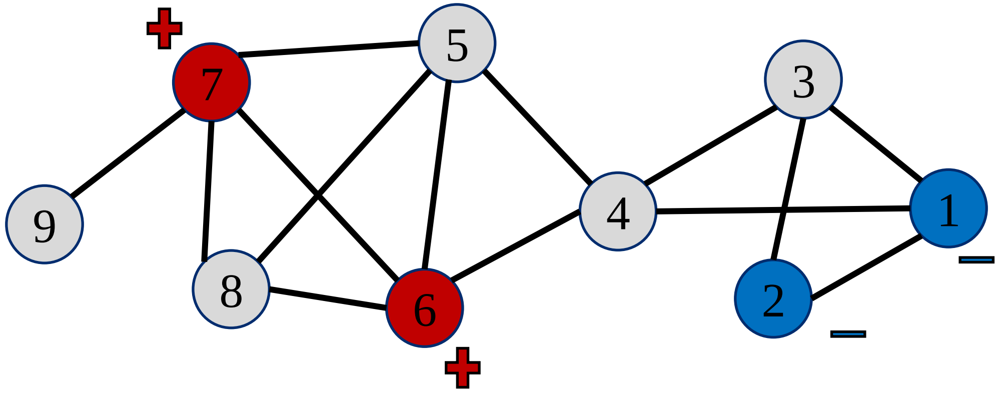
[1] Sen, Prithviraj, et al. "Collective classification in network data." *AI magazine* 29.3 (2008): 93-93.

Weighted-vote Relational Neighbor (wvRN)

- The labels of unlabeled node v_i are determined by a weighted vote among their neighbors $\mathcal{N}(v_i)$.

$$P(y_i = 1|\mathcal{N}(v_i)) = \frac{1}{|\mathcal{N}(v_i)|} \sum_{v_j \in \mathcal{N}(v_i)} P(y_j = 1|\mathcal{N}(v_j))$$

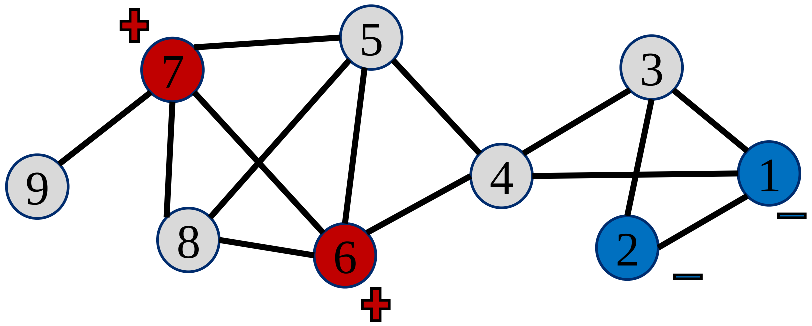
y_i is the label of node v_i



Macskassy, Sofus A., and Foster Provost. "A simple relational classifier." In Proceedings of the second workshop on multi-relational data mining (MRDM-2003) at KDD, pp. 64-76. 2003.

Weighted-vote Relational Neighbor (wvRN)

- The key idea
  1. Keep the labels of labeled nodes fixed.
  2. Recalculate probabilities iteratively based on neighbors' labels until convergence.



Macskassy, Sofus A., and Foster Provost. "A simple relational classifier." In Proceedings of the second workshop on multi-relational data mining (MRDM-2003) at KDD, pp. 64-76. 2003.

An Example of wvRN

- Initialization of unlabeled nodes

$$P(y_i = 1|\mathcal{N}(v_i)) = 0.5$$

  - For node v_3, $\mathcal{N}(v_3) = \{1, 2, 4\}$

$$P(y_1 = 1|\mathcal{N}(v_1)) = 0$$
$$P(y_2 = 1|\mathcal{N}(v_2)) = 0$$
$$P(y_4 = 1|\mathcal{N}(v_4)) = 0.5$$
$$P(y_3 = 1|\mathcal{N}(v_3)) = 1/3(0 + 0 + 0.5) = 0.17$$

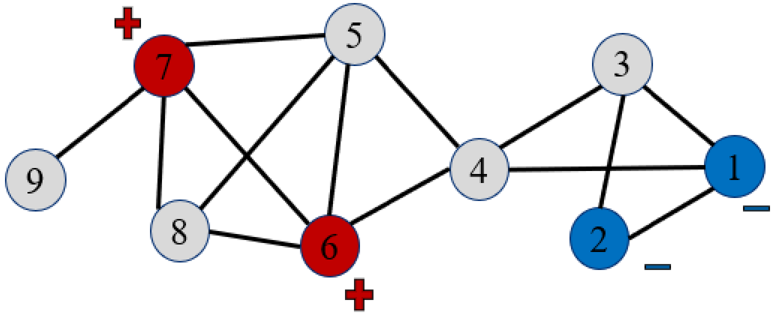
  - For node v_4, $\mathcal{N}(v_4) = \{1, 3, 5, 6\}$

$$P(y_4 = 1|\mathcal{N}(v_4)) = \frac{1}{4}(0 + 0.17 + 0.5 + 1) = 0.42$$

  - For node v_5, $\mathcal{N}(v_5) = \{4, 6, 7, 8\}$

$$P(y_5 = 1|\mathcal{N}(v_5)) = \frac{1}{4}(0.42 + 1 + 1 + 0.5) = 0.73$$

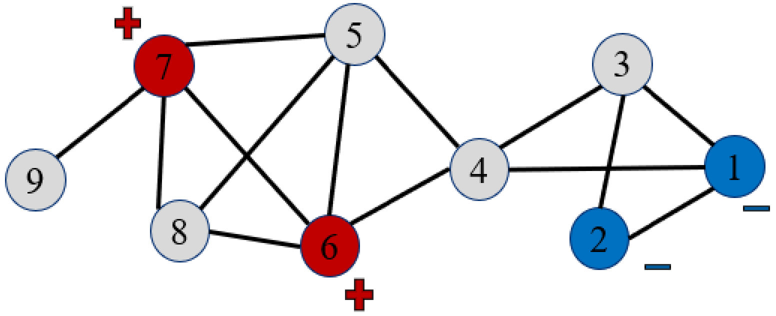
…



$$\mathbf{p}^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 0.17 \\ 0.42 \\ 0.73 \\ 1 \\ 1 \\ 0.91 \\ 1.00 \end{bmatrix}$$

Iteration Results

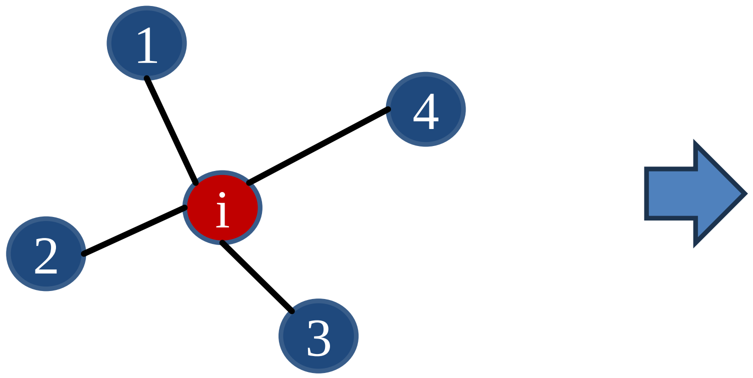
- Converge after 5 iterations
  - Nodes 5, 8, 9: + ($P(v_i) > 0.5$)
  - Node 3: - ($P(v_i) < 0.5$)
  - Node 4 is around $P(v_i) = 0.5$



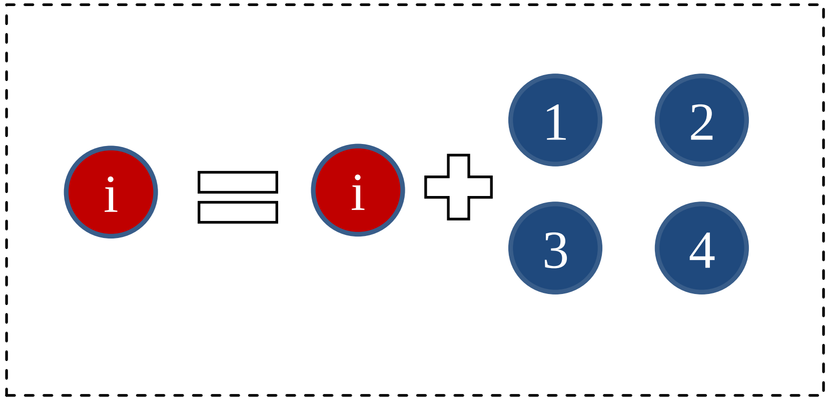
$$\mathbf{p}^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 0.17 \\ 0.42 \\ 0.73 \\ 1 \\ 1 \\ 0.91 \\ 1.00 \end{bmatrix}, \mathbf{p}^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 0.14 \\ 0.47 \\ 0.85 \\ 1 \\ 1 \\ 0.95 \\ 1.00 \end{bmatrix}, \mathbf{p}^{(3)} = \begin{bmatrix} 0 \\ 0 \\ 0.16 \\ 0.50 \\ 0.86 \\ 1 \\ 1 \\ 0.95 \\ 1.00 \end{bmatrix}, \mathbf{p}^{(4)} = \begin{bmatrix} 0 \\ 0 \\ 0.17 \\ 0.51 \\ 0.87 \\ 1 \\ 1 \\ 0.96 \\ 1.00 \end{bmatrix}, \mathbf{p}^{(5)} = \begin{bmatrix} 0 \\ 0 \\ 0.17 \\ 0.51 \\ 0.87 \\ 1 \\ 1 \\ 0.96 \\ 1.00 \end{bmatrix}.$$

Label Propagation (Consistency Method)

- Intuition
  1. Each node iteratively spreads its label to neighbors.
  2. Each node partly keeps its initial label information.



Given graph and label information



The process of label propagation

- Zhou D, Bousquet O, Lal T N, et al. Learning with local and global consistency[C] //Advances in neural information processing systems. 2004: 321-328.

Symbol Notations

- **G**: Input network structure
- **F**: Classification result matrix on nodes

$$F_{n\times c} = \begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1c} \\ \cdots & \cdots & \cdots\cdot & \cdots\cdot \\ F_{n1} & F_{n2} & \cdots & F_{nc} \end{bmatrix} \quad \text{by labeling } v_i \text{ as } y_i = \operatorname{argmax}_{j\le c} F_{ij}$$

- **Y**: Initial labels of nodes

$$Y_{ij} = \begin{cases} 1, & \text{if } v_i \text{ is labeled as } y_i = j \\ 0, & \text{otherwise} \end{cases}$$

Consistency Method

- Symmetric normalization of G:

$$S = D^{-\frac{1}{2}} G D^{-\frac{1}{2}}$$

  where $D_{ii} = \sum_j G_{ij}$ is the degree matrix of the network.

- Iterate $F(t+1) = \alpha S F(t) + (1-\alpha) Y$ until converge

- Let F^* represent the limit of the sequence $\{F(t)\}$. The final classification result is:

$$y_i = \arg\max_{j \leq c} F^*_{ij}$$

The Closed-Form Solution of the Consistency Method

- Through the iterative equation, we obtain:

$$F(t) = (\alpha S)^{t-1}Y + (1-\alpha)\sum_{i=0}^{t-1}(\alpha S)^i Y.$$

Since $0 < \alpha < 1$ and the $\boxed{\text{eigenvalues of } S \text{ in [-1, 1]}}$

The advantage of normalization

- As $\lim_{t\to\infty}(\alpha S)^{t-1} = 0,$ and $\lim_{t\to\infty}\sum_{i=0}^{t-1}(\alpha S)^i = (I-\alpha S)^{-1}$, we get:

$$F^* = \lim_{t\to\infty} F(t) = (1-\alpha)(I-\alpha S)^{-1}Y$$

Thanks for your time.
QA.