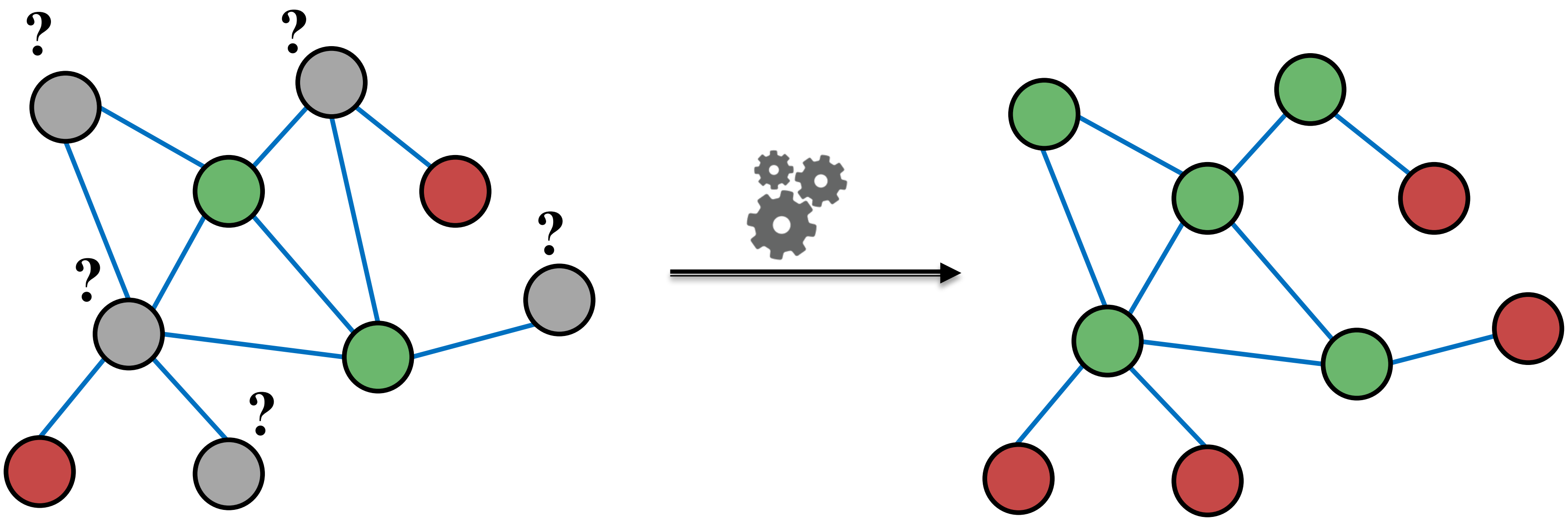
Stanford CS224W: Message Passing and Node Classification

CS224W: Machine Learning with Graphs
Jure Leskovec, Stanford University
http://cs224w.stanford.edu

Today's Lecture: Outline

- **Main question today:** Given a network with labels on some nodes, how do we assign labels to all other nodes in the network?
- **Example:** In a network, some nodes are fraudsters, and some other nodes are fully trusted. **How do you find the other fraudsters and trustworthy nodes?**
- We already discussed node embeddings as a method to solve this in Lecture 3

Example: Node Classification



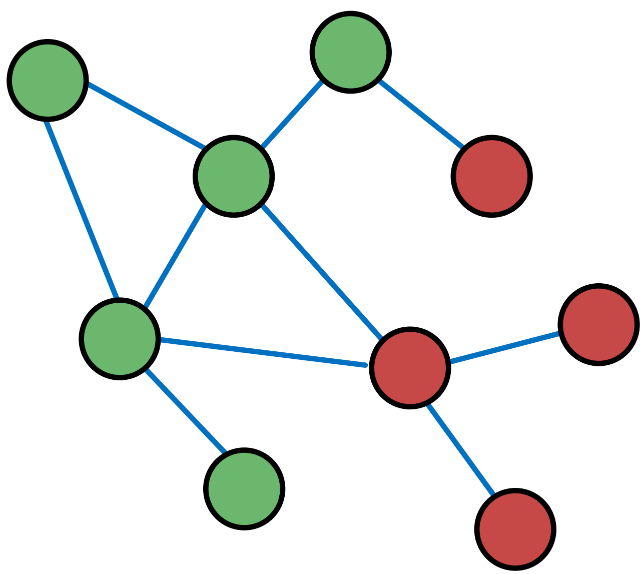
- Given labels of some nodes
- Let's predict labels of unlabeled nodes
- This is called **semi-supervised node classification**

Today's Lecture: Outline

- **Main question today:** Given a network with labels on some nodes, how do we assign labels to all other nodes in the network?
- **Today we will discuss an alternative framework: Message passing**
- **Intuition:** **Correlations (dependencies)** exist in networks.
  - In other words: Similar nodes are connected.
  - **Key concept** is **collective classification**: Idea of assigning labels to all nodes in a network together.
- **We will look at three techniques today:**
  - **Relational classification**
  - **Iterative classification**
  - **Correct & Smooth**

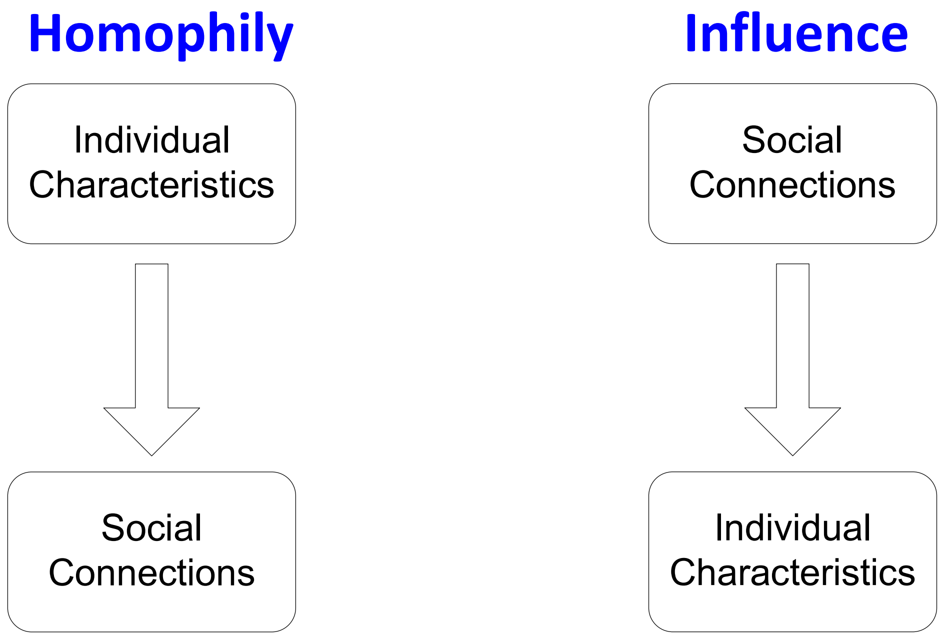
Correlations Exist in Networks

- Behaviors of nodes are **correlated** across the links of the network
- **Correlation**: Nearby nodes have the same color (belonging to the same class)



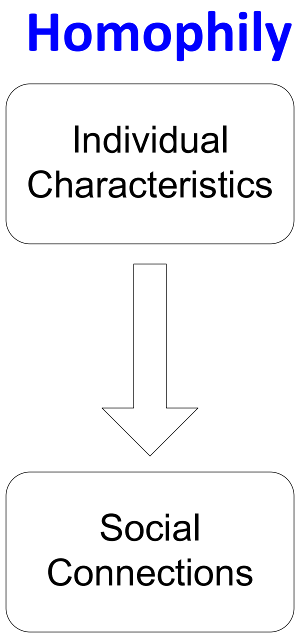
Correlations Exist in Networks

- **Two explanations for why behaviors of nodes in networks are correlated:**



Social Homophily

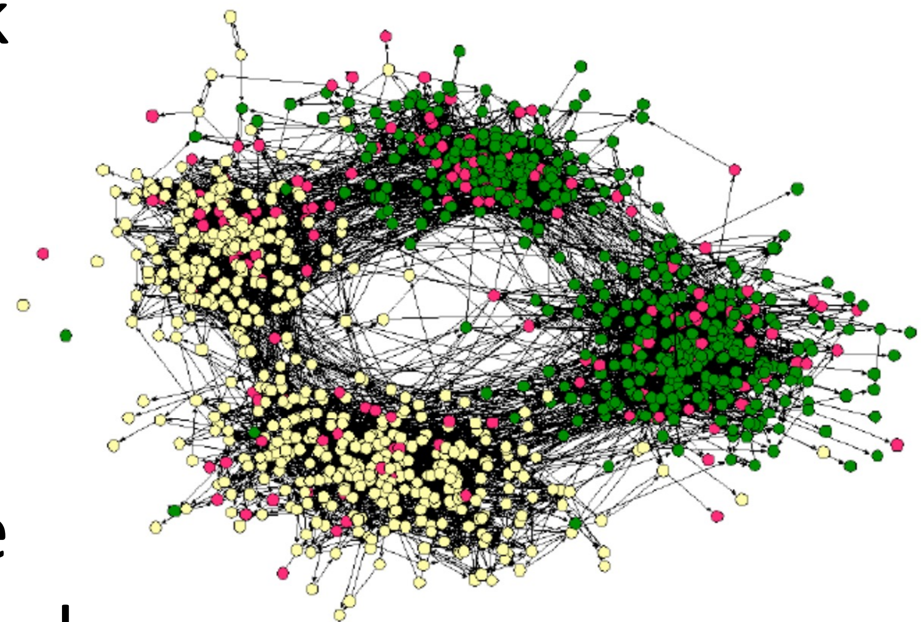
- **Homophily**: The tendency of individuals to associate and bond with similar others
  - *"Birds of a feather flock together"*
  - It has been observed in a vast array of network studies, based on a variety of attributes (e.g., age, gender, organizational role, etc.)
  - **Example**: Researchers who focus on the same research area are more likely to establish a connection (meeting at conferences, interacting in academic talks, etc.)



Homophily: Example

Example of homophily

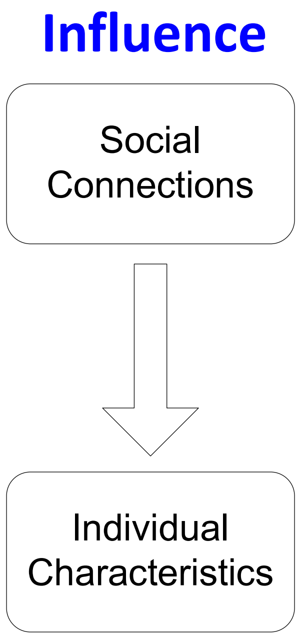
- Online social network
  - Nodes = people
  - Edges = friendship
  - Node color = interests (sports, arts, etc.)
- People with the same interest are more closely connected due to homophily



(Easley and Kleinberg, 2010)

Social Influence: Example

- **Influence**: Social connections can influence the individual characteristics of a person.
  - **Example**: I recommend my musical preferences to my friends, until one of them grows to like my same favorite genres!



Stanford CS224W: How do we leverage node correlations in networks?

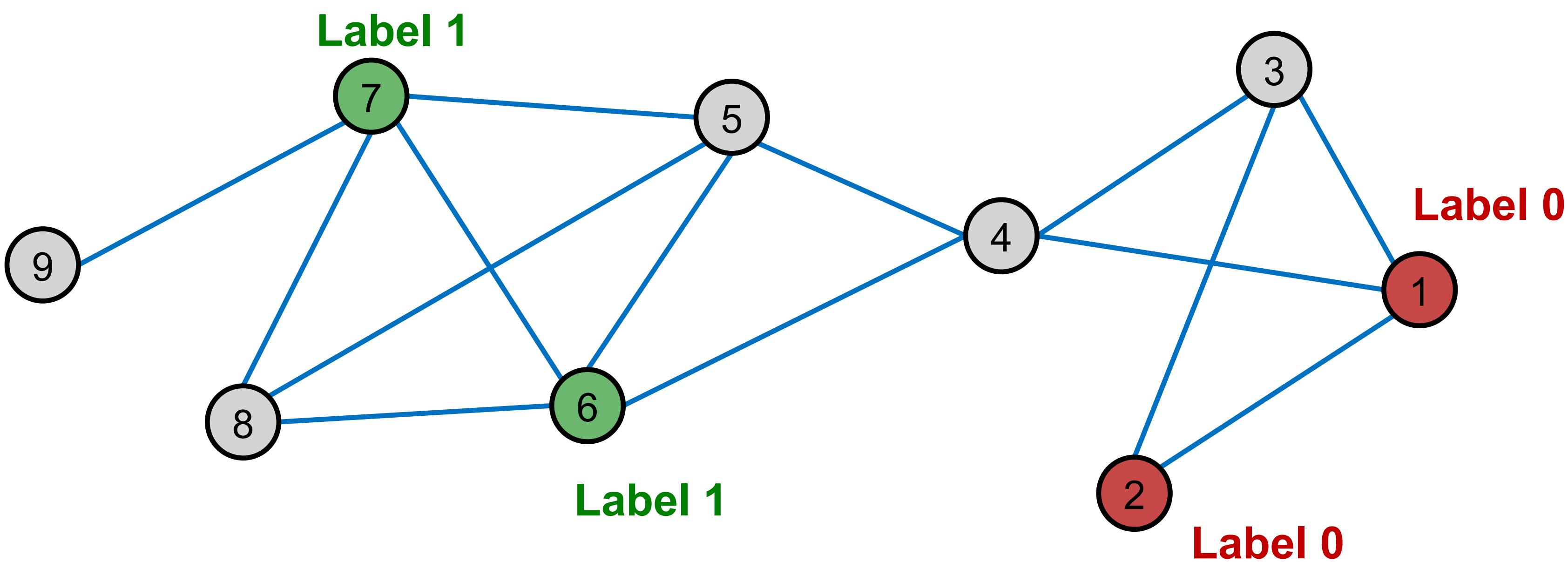
CS224W: Machine Learning with Graphs

Jure Leskovec, Stanford University

http://cs224w.stanford.edu

Classification with Network Data

- **How do we leverage this correlation observed in networks to help predict node labels?**



How do we predict the labels for the nodes in grey?

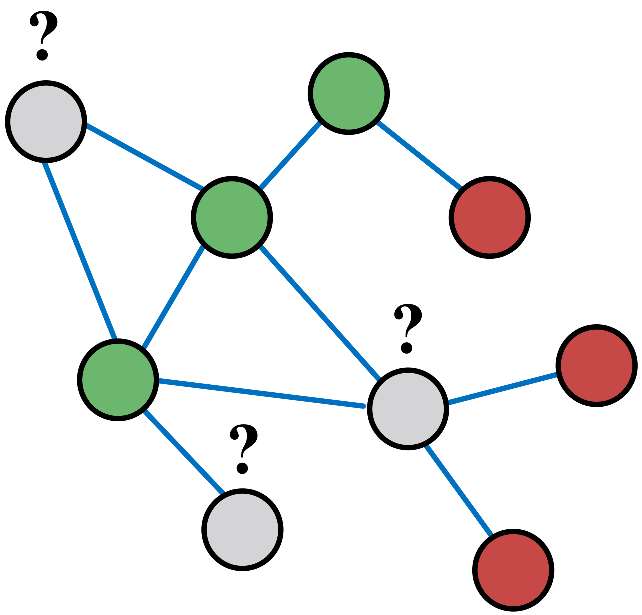
Motivation (1)

- **Similar nodes are typically close together or directly connected in the network:**
  - **Guilt-by-association**: If I am connected to a node with label X, then I am likely to have label X as well.
  - **Example: Malicious/benign web page:** Malicious web pages link to one another to increase visibility, look credible, and rank higher in search engines

Motivation (2)

- **Classification label** of a node v in network may depend on:
  - Features of v
  - Labels of the nodes in v's neighborhood
  - Features of the nodes in v's neighborhood

Semi-supervised Learning (1)



Formal setting:

Given:
- Graph
- Few labeled nodes

Find: Class (red/green) of remaining nodes

Main assumption: There is homophily in the network

Semi-supervised Learning (2)

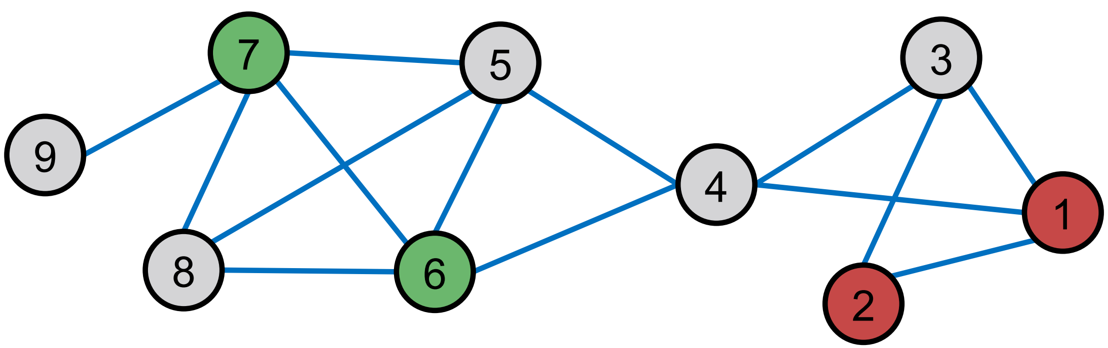
Example task:

- Let $\boldsymbol{A}$ be a $n \times n$ adjacency matrix over n nodes
- Let $\mathrm{Y} = \{0, 1\}^n$ be a vector of **labels**:
  - $\mathrm{Y}_v = 1$ belongs to Class 1
  - $\mathrm{Y}_v = 0$ belongs to Class 0
  - There are unlabeled node needs to be classified
- **Goal:** Predict which **unlabeled** nodes are likely **Class 1**, and which are likely **Class 0**

Problem Setting

- **How to predict the labels Y_v for the unlabeled nodes v (in grey color)?**
- Each node v has a feature vector f_v
- Labels for some nodes are given (1 for green, 0 for red)
- **Task:** Find $P(Y_v)$ given all features and the network

$P(Y_v) = ?$



Example applications:

- **Many applications under this setting:**
  - Document classification
  - Part of speech tagging
  - Link prediction
  - Optical character recognition
  - Image/3D data segmentation
  - Entity resolution in sensor networks
  - Spam and fraud detection

Overview of What is Coming

- We focus on **semi-supervised binary node classification**
- **We will introduce three approaches:**
  - **Relational classification**
  - **Iterative classification**
  - **Correct & Smooth**

Stanford CS224W: Relational Classification

CS224W: Machine Learning with Graphs

Jure Leskovec, Stanford University

http://cs224w.stanford.edu

Probabilistic Relational Classifier (1)

- **Idea:** **Propagate node labels across the network**
  - Class probability Y_v of node v is a weighted average of class probabilities of its neighbors.
- For **labeled nodes** v, initialize label Y_v with ground-truth label Y_v^*.
- For **unlabeled nodes**, initialize $Y_v = 0.5$.
- **Update** all nodes in a random order until convergence or until maximum number of iterations is reached.

Stanford CS224W: Iterative Classification

CS224W: Machine Learning with Graphs

Jure Leskovec, Stanford University

http://cs224w.stanford.edu

Iterative Classification

- **Relational classifier does not use node attributes.**
- **How can one leverage them?**
- **Main idea of iterative classification:** Classify node v based on its **attributes $\boldsymbol{f_v}$** as well as **labels $\boldsymbol{z_v}$** of neighbor set N_v.

Summary

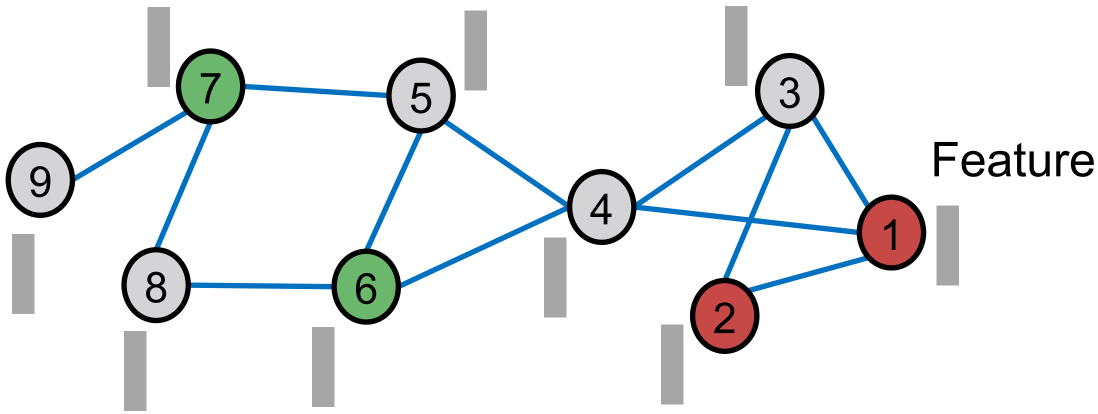
- **We talked about 2 approaches to collective classification**
- **Relational classification**
  - Iteratively update probabilities of node belonging to a label class based on its neighbors
- **Iterative classification**
  - Improve over collective classification to handle attribute/feature information
  - Classify node v based on its **features** as well as **labels** of neighbors

Stanford CS224W: Collective Classification: Correct & Smooth

CS224W: Machine Learning with Graphs
Jure Leskovec, Stanford University
http://cs224w.stanford.edu

Correct & Smooth

- **Setting:** A partially labeled graph and features over nodes.



- **C&S follows the three-step procedure:**
  1. Train base predictor
  2. Use the base predictor to predict soft labels of all nodes.
  3. **Post-process the predictions using graph structure** to obtain the final predictions of all nodes.

Correct & Smooth: Summary

- Correct & Smooth (C&S) **uses graph structure to post-process** the soft node labels predicted by any base model.
- **Correction step**: Diffuse and correct for the training errors of the base predictor.
- **Smooth step**: Smoothen the prediction of the base predictor.
- C&S achieves strong performance on semi-supervised node classification.

Summary

- **We learned how to leverage correlation in graphs to make prediction on nodes.**
- **Key techniques:**
  - Relational classification
  - Iterative classification
  - Correct & Smooth