# Message Passing and Node Classification

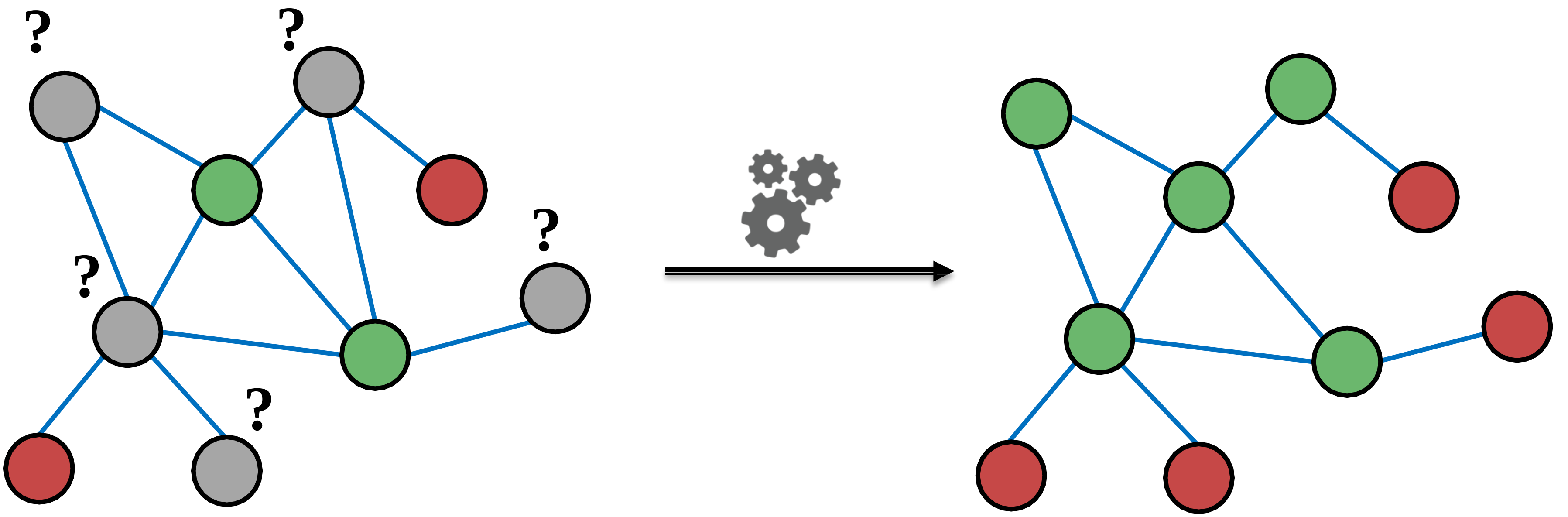
# Outline

- **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

# Example: Node Classification



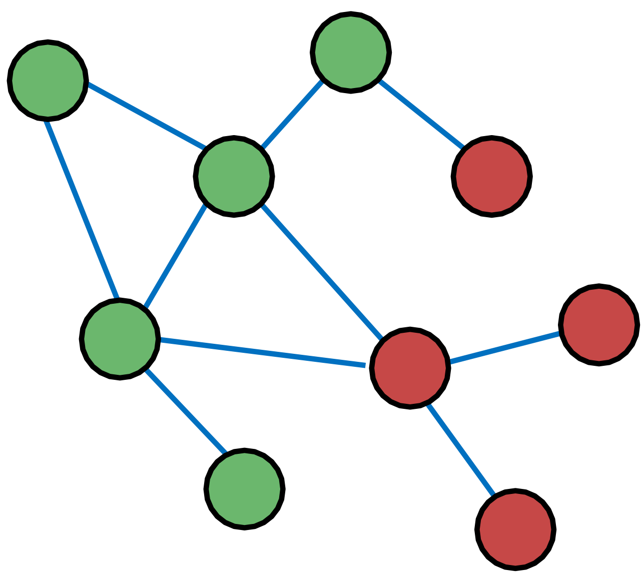

- Given labels of some nodes
- Let’s predict labels of unlabeled nodes
- This is called semi-supervised node classification

# Outline

- **Today we will discuss some intuitions behind the framework: message passing**
- **Intuition: Correlations** 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 some techniques today:**
  - **Relational classification**
  - **Iterative classification**

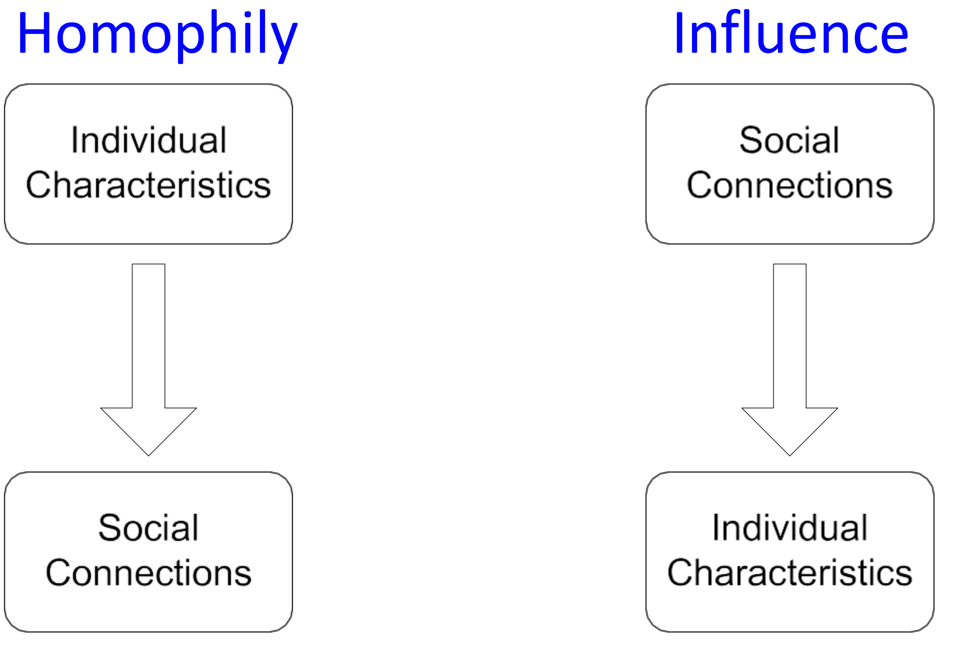
# Correlations in Networks

- Individual behaviors are **correlated** in the network
- **Correlation**: nearby nodes have the same color (belonging to the same class)



# Correlations in Networks

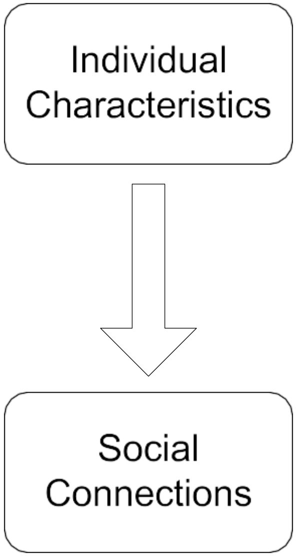
- **Main types of dependencies that lead to correlation:**



# 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.)

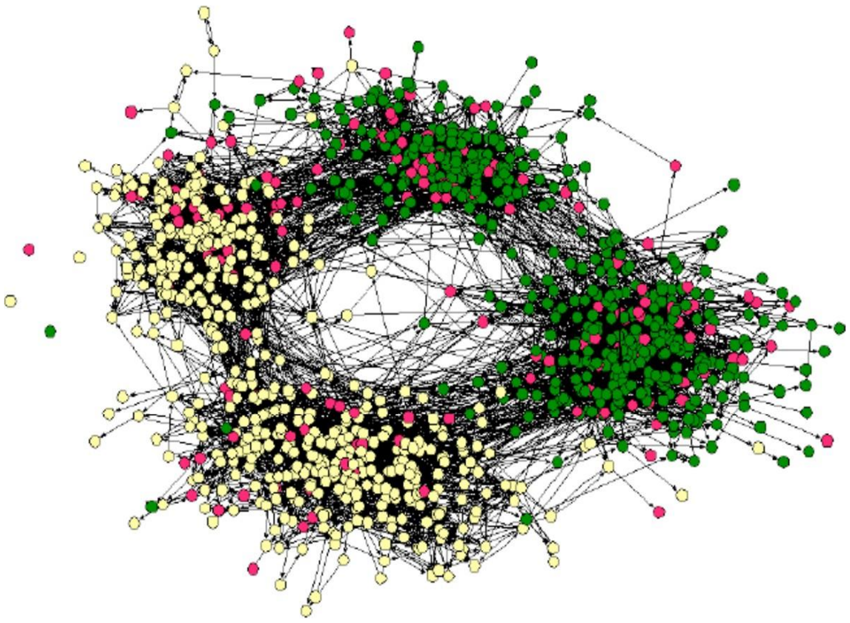
**Homophily**



# Homophily: Example

**Example of homophily**

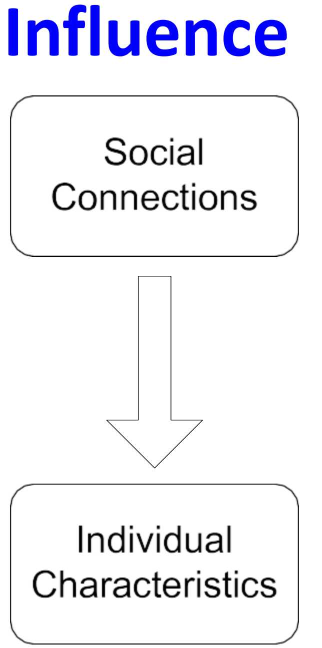
- Online social network
  - Nodes = people
  - Edges = friendship
  - Node color = interests (sports, arts, etc.)
- People with the same interest are more closely connected due to homophily



(Easley and Kleinberg, 2010)

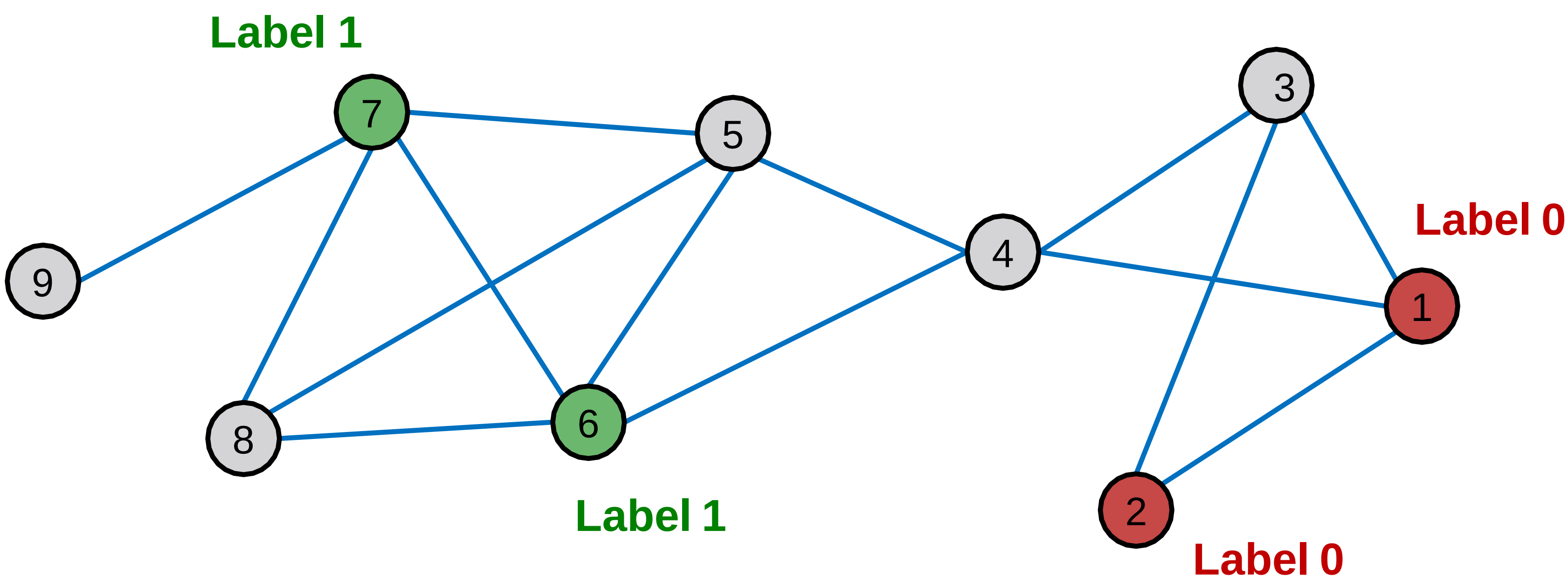
# Influence

- **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!



# Classification with Networks

- 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?

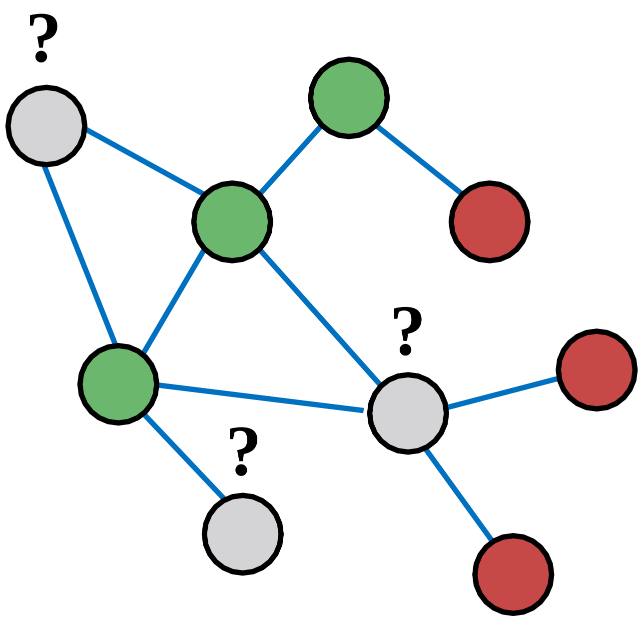
# Motivation

- **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

- **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




**Formal setting**

**Given**:

- Graph
- Few labelled nodes

**Find**: class (red/green) of remaining nodes

**Main assumption**: There is homophily in the network

# Semi-supervised Learning

**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**

# Collective Classification

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

# Collective Classification

- **Intuition:** Simultaneous classification of interlinked nodes using correlations
- Probabilistic framework
- Markov Assumption: *the label* $Y_v$ *of one node* $v$ *depends on the labels of its neighbors* $N_v$

$$P(Y_v) = P(Y_v | N_v)$$

- Collective classification involves 3 steps:

| Local Classifier | Relational Classifier | Collective Inference |
|---|---|---|
| • Assign initial labels | • Capture correlations between nodes | • Propagate correlations through network |

# Collective Classification

**Local Classifier**

- Assign initial labels

**Relational Classifier**

- Capture correlations between nodes

**Collective Inference**

- Propagate correlations through network

Local Classifier: Used for initial label assignment

- Predicts label based on node attributes/features
- Standard classification task
- Does not use network information

Relational Classifier: Capture correlations

- Learns a classifier to label one node based on the labels and/or attributes of its neighbors
- This is where network information is used

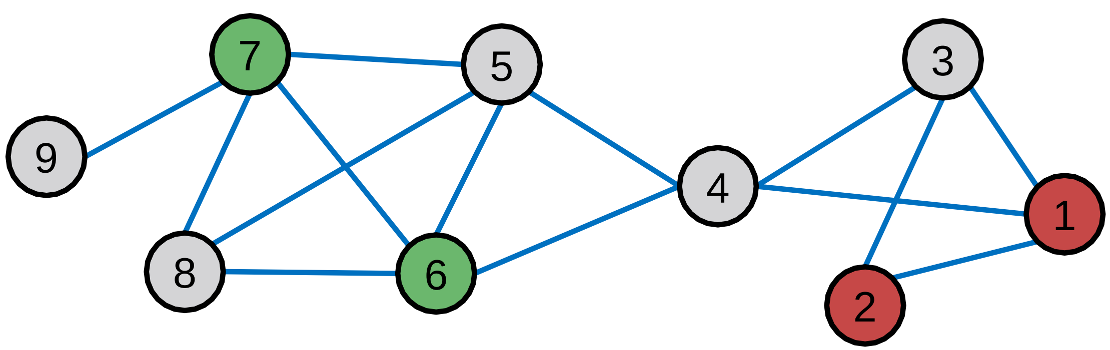
Collective Inference: Propagate the correlation

- Apply relational classifier to each node iteratively
- Iterate until the inconsistency between neighboring labels is minimized
- Network structure affects the final prediction

# 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 $P(Yv)$ given all features and the network

$P(Yv) = ?$



# What next?

- We focus on semi-supervised node classification
- Intuition is based on **homophily**: Similar nodes are typically close together or directly connected
- **Techniques we will introduce:**
  - **Relational classification**
  - **Iterative classification**

# Relation Classification and Iterative Classification

# Collective Classification Models

- **Relational classifiers**
- Iterative classification

# Probabilistic Relational Classifier

- **Basic idea:** 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

# Probabilistic Relational Classifier

- **Update** for each node $v$ and label $c$ (e.g. 0 or 1)

$$P(Y_v = c) = \frac{1}{\sum_{(v,u)\in E} A_{v,u}} \sum_{(v,u)\in E} A_{v,u} P(Y_u = c)$$

  - If edges have strength/weight information, $A_{v,u}$ can be the edge weight between $v$ and $u$
  - $P(Y_v = c)$ is the probability of node $v$ having label $c$

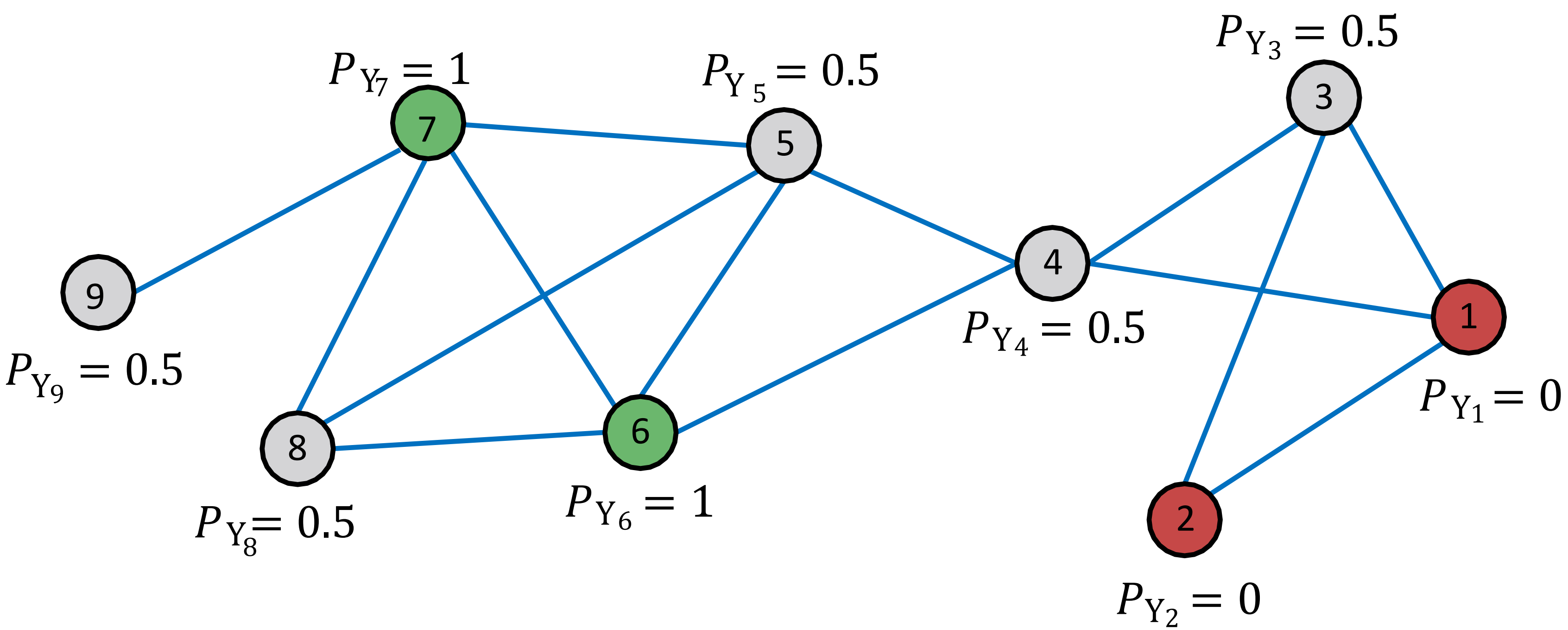
- Challenges:
  - Convergence is not guaranteed
  - Model cannot use node feature information

# Initialization

**Initialization**:

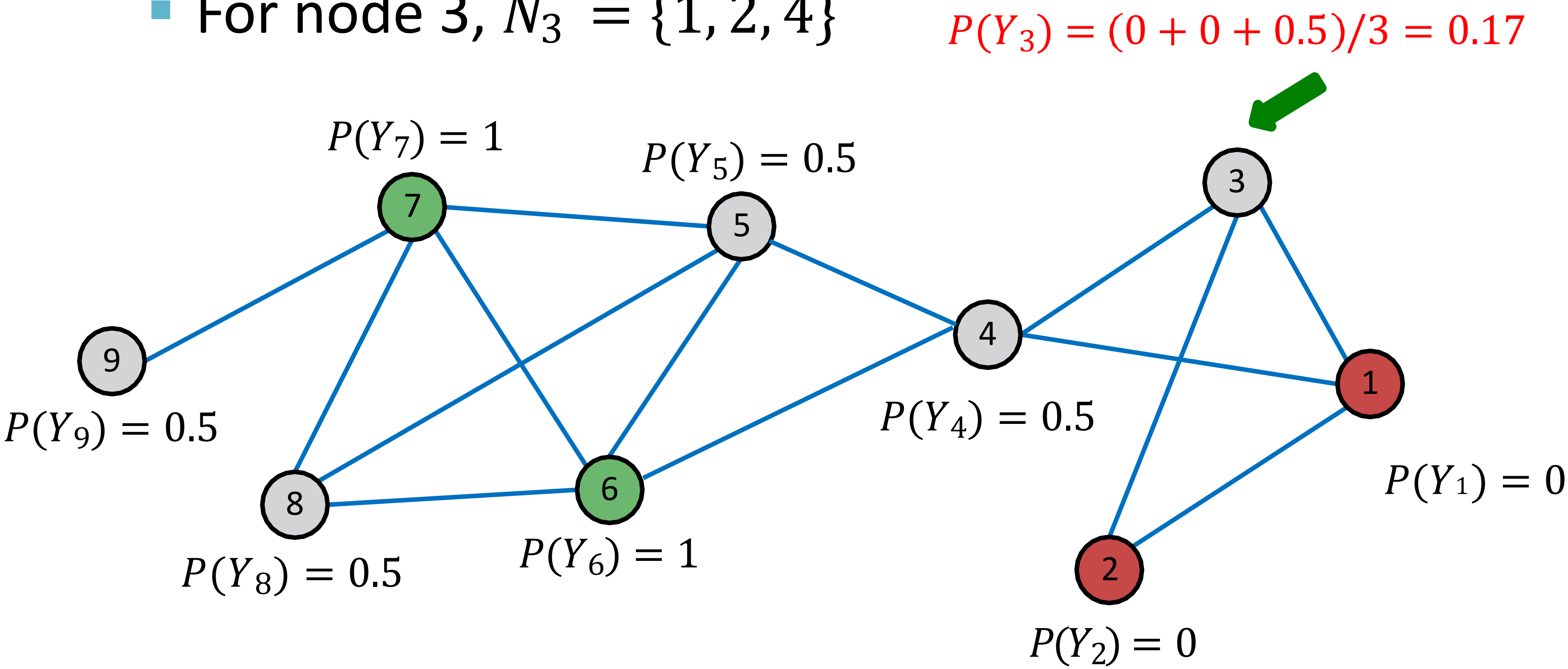
- All labeled nodes with their labels
- All unlabeled nodes 0.5 (belonging to class 1 with probability 0.5)

Let $P_{Y_1} = P(Y_1 = 1)$



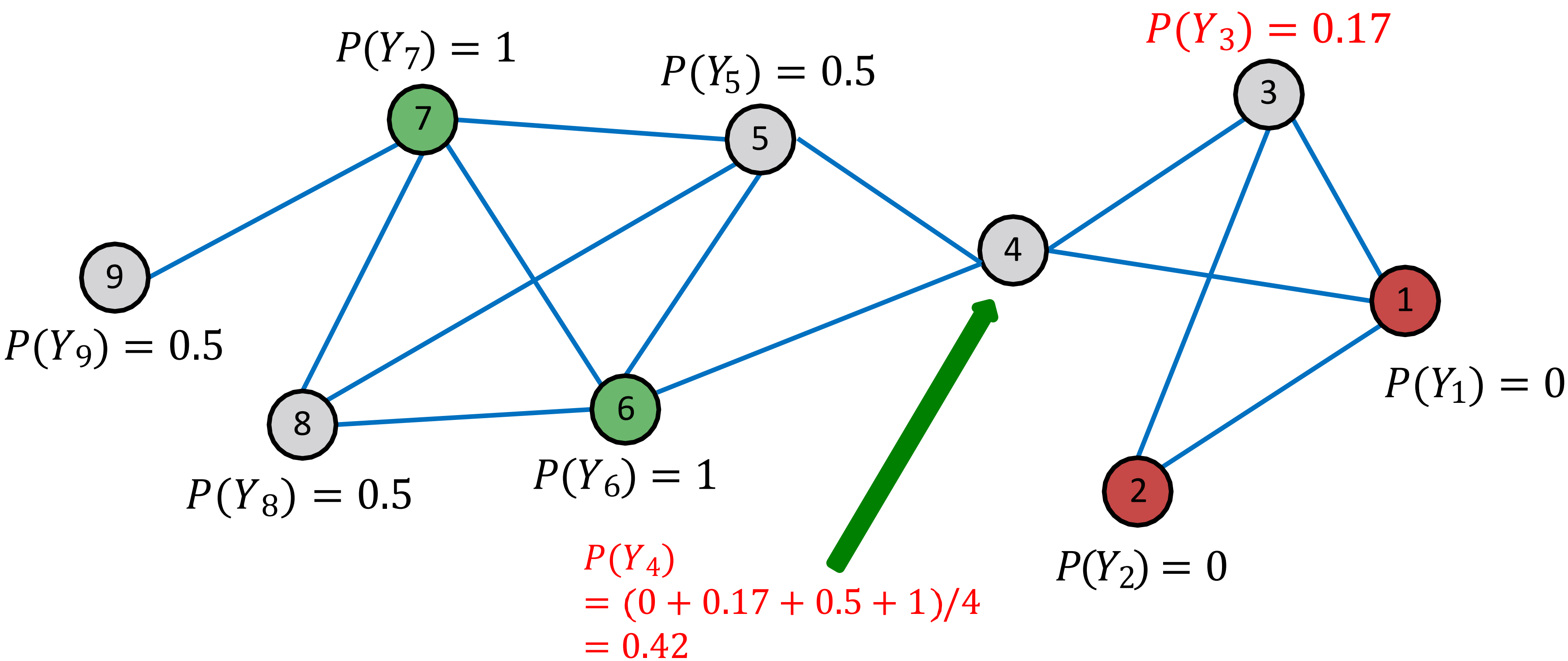
# $1^{st}$ Iteration, Update Node 3

- Update for the $1^{st}$ Iteration:
  - For node 3, $N_3 = \{1, 2, 4\}$



# $1^{st}$ Iteration, Update Node 4

- Update for the $1^{st}$ Iteration:
  - For node 4, $N_4 = \{1, 3, 5, 6\}$



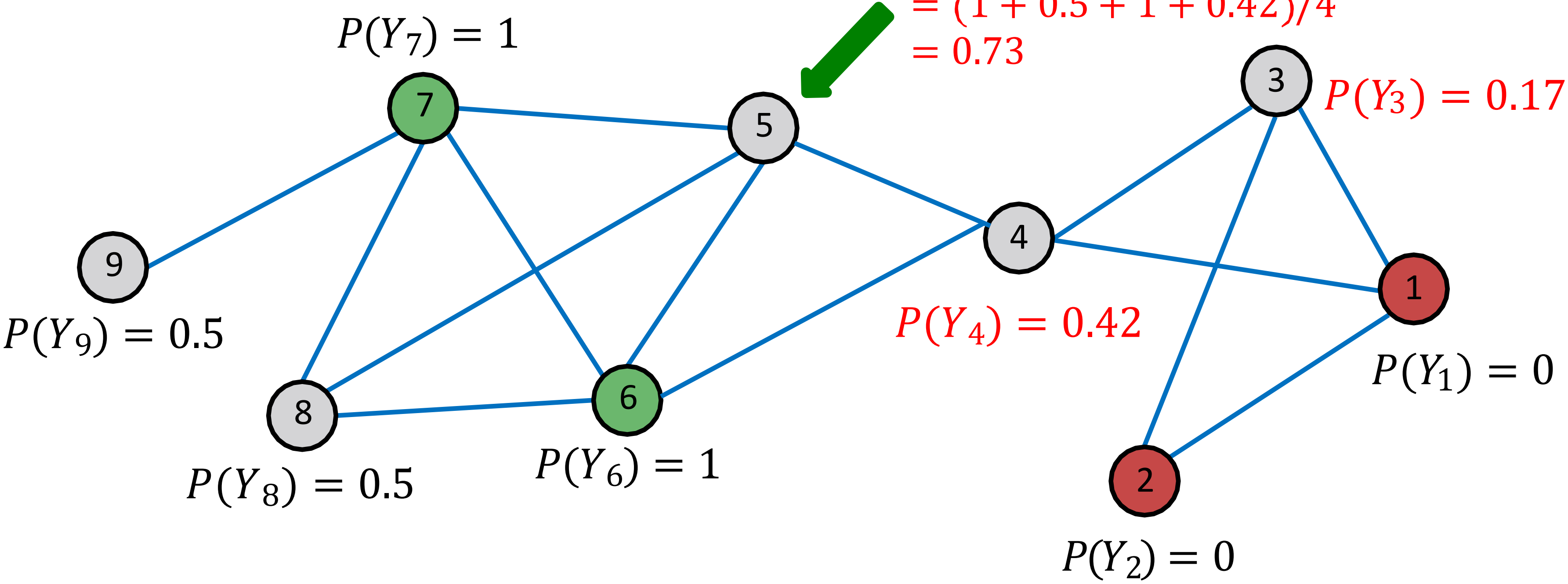

After Node 3 is updated

# $1^{st}$ Iteration, Update Node 5

- Update for the $1^{st}$ Iteration:
  - For node 5, $N_5 = \{4, 6, 7, 8\}$

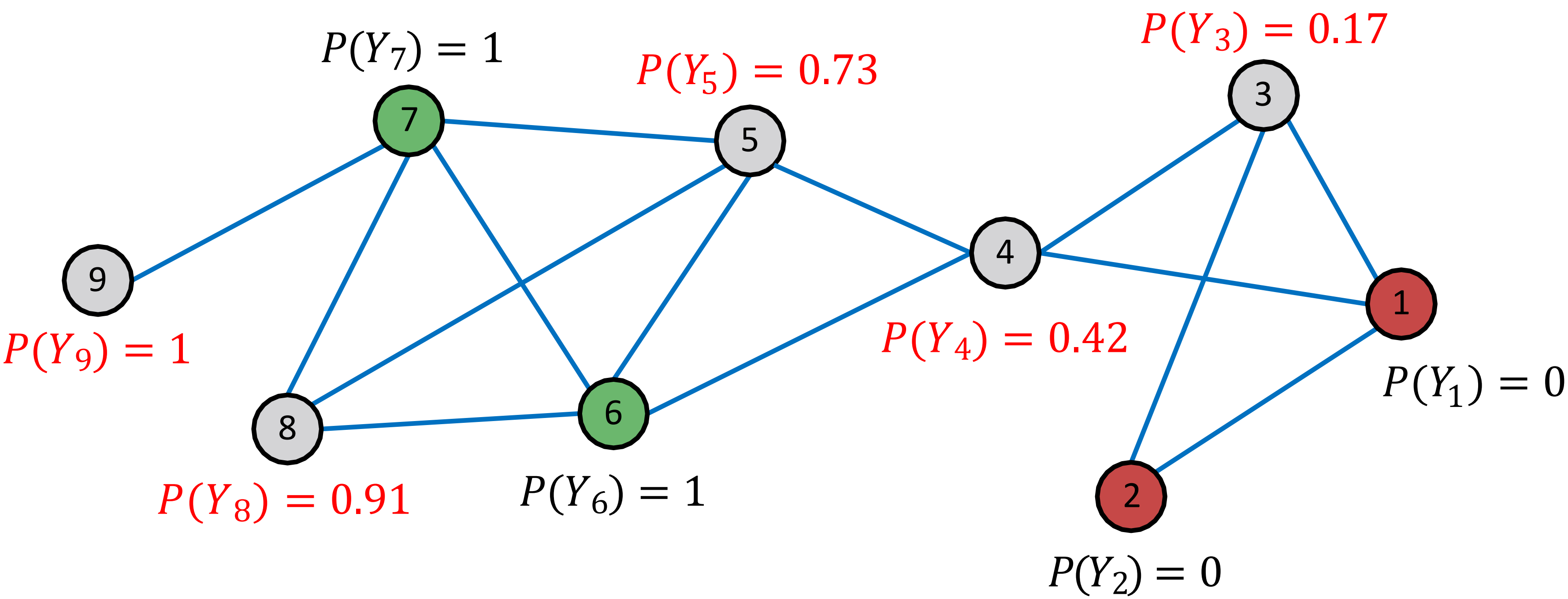
After nodes 3 and 4 are updated

$P(Y_5)$
$= (1 + 0.5 + 1 + 0.42)/4$
$= 0.73$



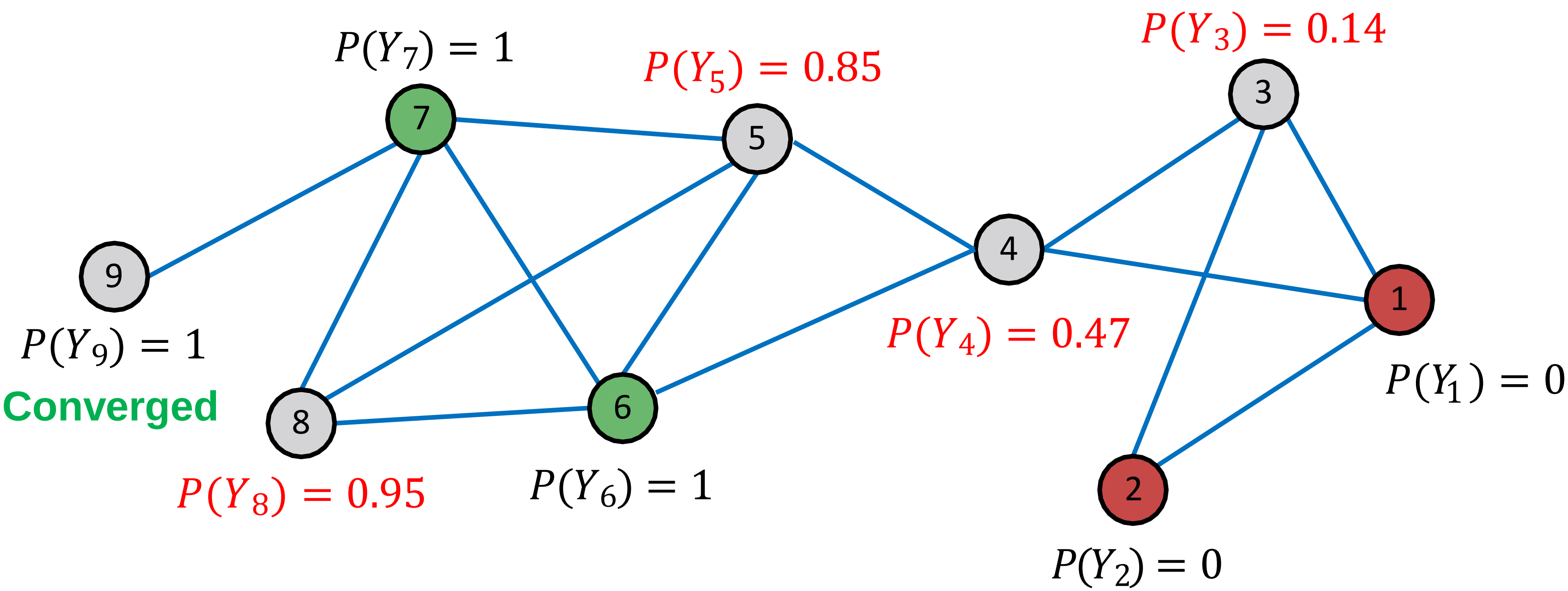
# End of 1st Iteration

- After Iteration 1
  (a round of updates for all unlabeled nodes)



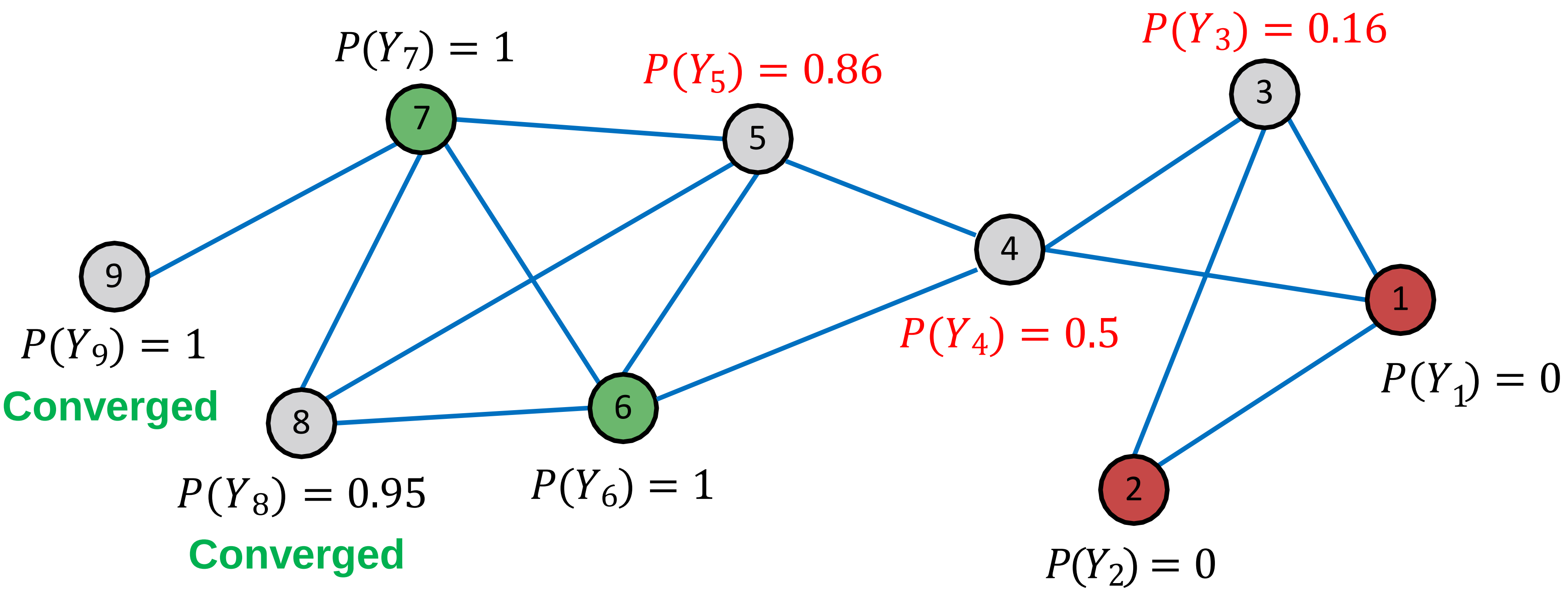
# After 2nd Iteration

- After Iteration 2



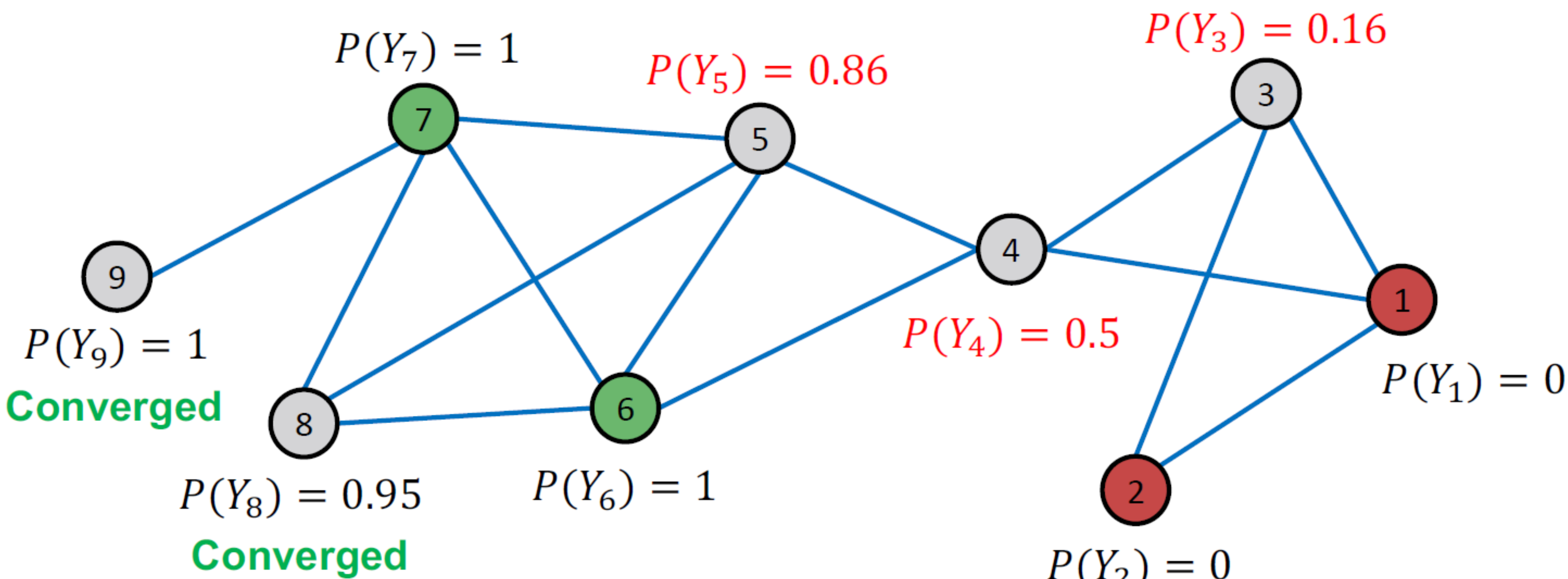
# After 3rd Iteration

- After Iteration 3



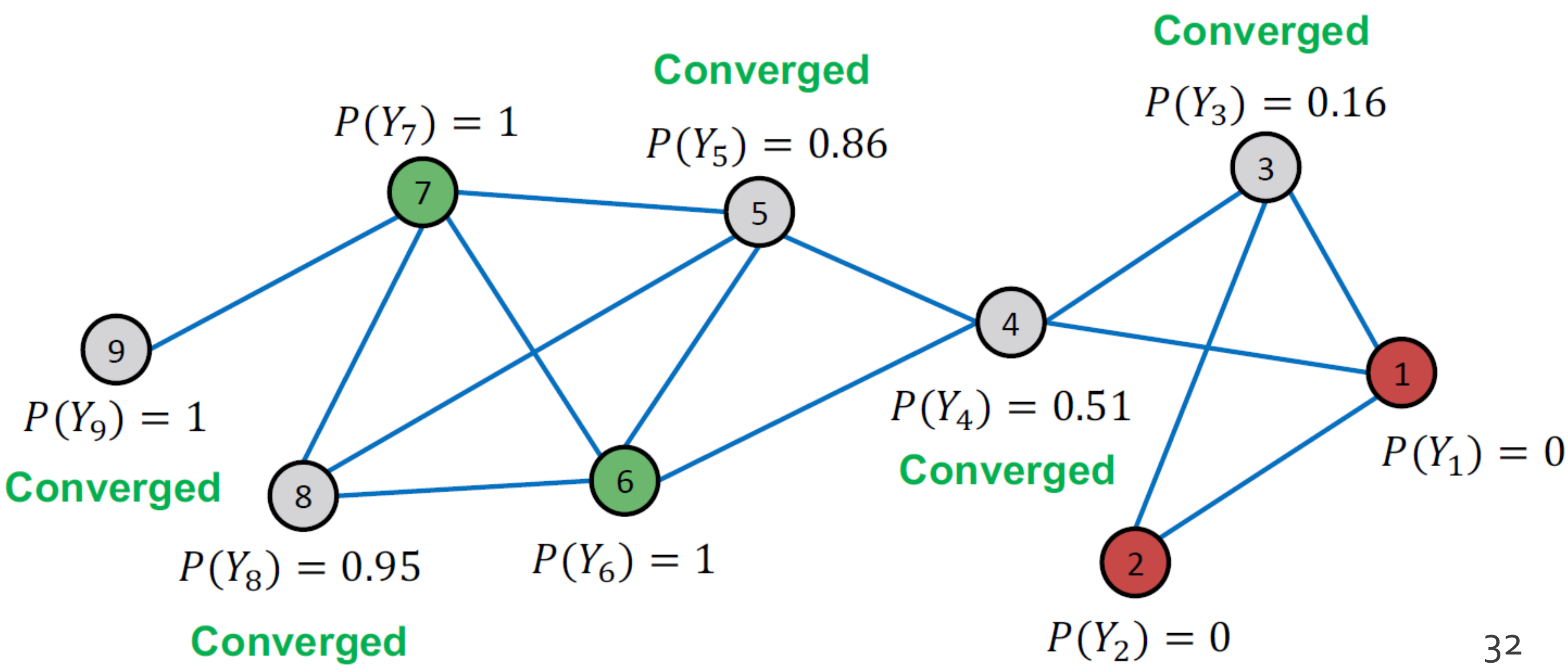
# After $4^{th}$ Iteration

- After Iteration 4



# Convergence

- All scores stabilize after 4 iterations. We therefore predict:
  - **Nodes 4, 5, 8, 9 belong to class 1** ($P_{Y_v} > 0.5$)
  - **Nodes 3 belong to class 0** ($P_{Y_v} < 0.5$)



# Collective Classification Models

- Relational classifiers
- **Iterative classification**

# Iterative Classification

- **Relational classifiers do 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 $\boldsymbol{N_v}$

# Iterative Classification

- **Input: Graph**
  - $f_v$ : feature vector for node $v$
  - Some nodes $v$ are labeled with $Y_v$
- **Task:** Predict label of unlabelled nodes

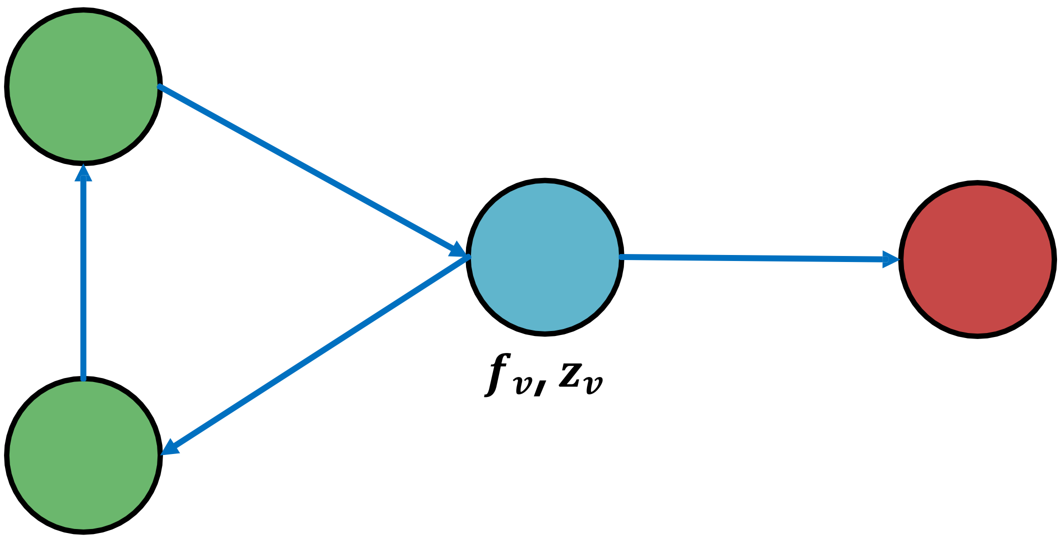
- **Approach: Train two classifiers:**
- $\phi_1(f_v)$ = Predict node label based on node feature vector $f_v$
- $\phi_2(f_v, z_v)$ = Predict label based on node feature vector $f_v$ and summary $z_v$ of labels of $v$'s neighbors.

# Computing Summary $\boldsymbol{z_v}$

**How do we compute the summary $z_v$ of labels of $v$'s neighbors $N_v$?**

Ideas: $\boldsymbol{z_v}$ **= vector captures labels around node $v$**

- Histogram of the number (or fraction) of each label in $N_v$
- Most common label in $N_v$
- Number of different labels in $N_v$



# Architecture of Iterative Classifiers

- **Phase 1: Classify based on node attributes alone**
  - On a **training set**, train classifier (e.g., linear classifier, neural networks, …):
  - $\phi_1(f_v)$ to predict $Y_v$ based on $f_v$
  - $\phi_2\ (f_v, z_v)$ to predict $Y_v$ based on $f_v$ and summary $z_v$ of labels of $v$'s neighbors

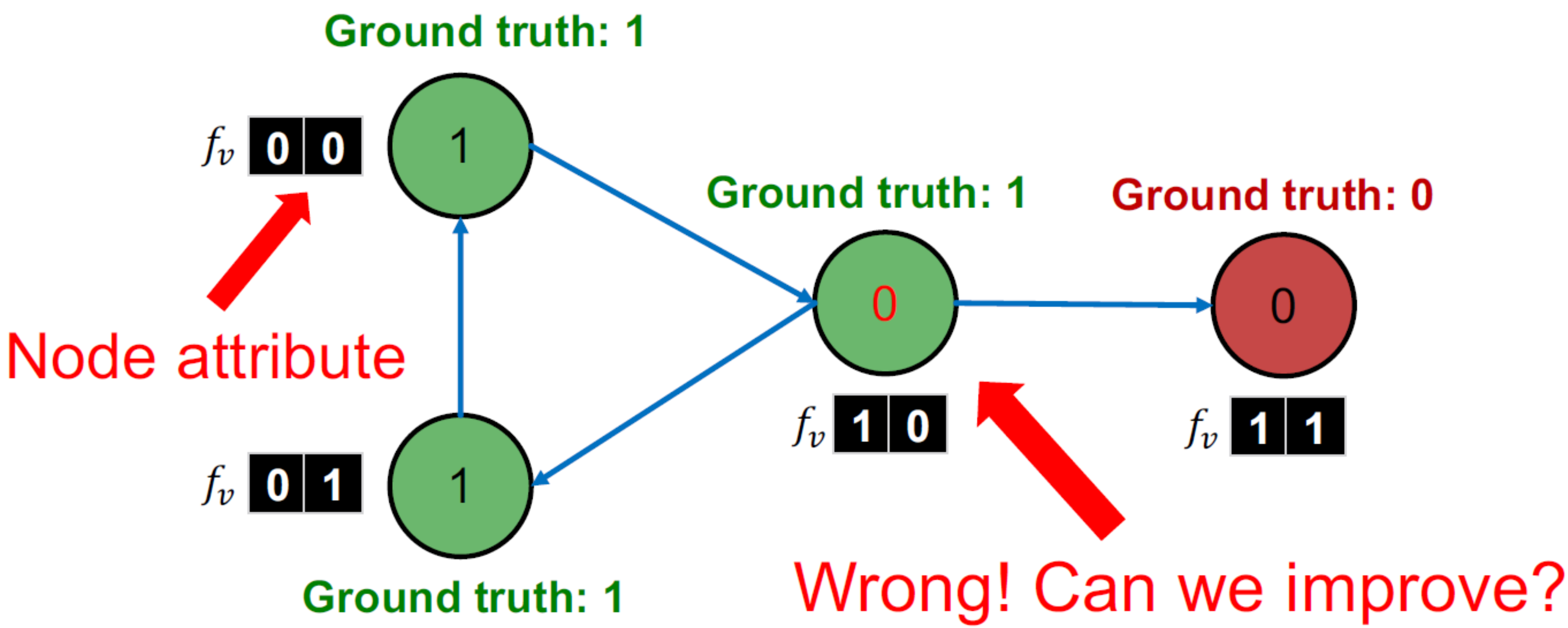
- **Phase 2: Iterate till convergence**
  - On **test set**, set labels $Y_v$ based on the classifier $\phi_1$, compute $z_v$ and predict the labels with $\phi_2$
  - **Repeat** for each node $v$
    - Update $z_v$ based on $Y_u$ for all $u \in N_v$
    - Update $Y_v$ based on the new $z_v$ ($\phi_2$)
  - Iterate until class labels stabilize or max number of iterations is reached
  - Note: Convergence is not guaranteed

# Example: Web Page Classification

- **Input:** Graph of web pages
- **Node:** Web page
- **Edge:** Hyper-link between web pages
  - **Directed edge:** a page points to another page
- **Node features:** Webpage description
  - For simplicity, we only consider 2 binary features
- **Task:** Predict the topic of the webpage

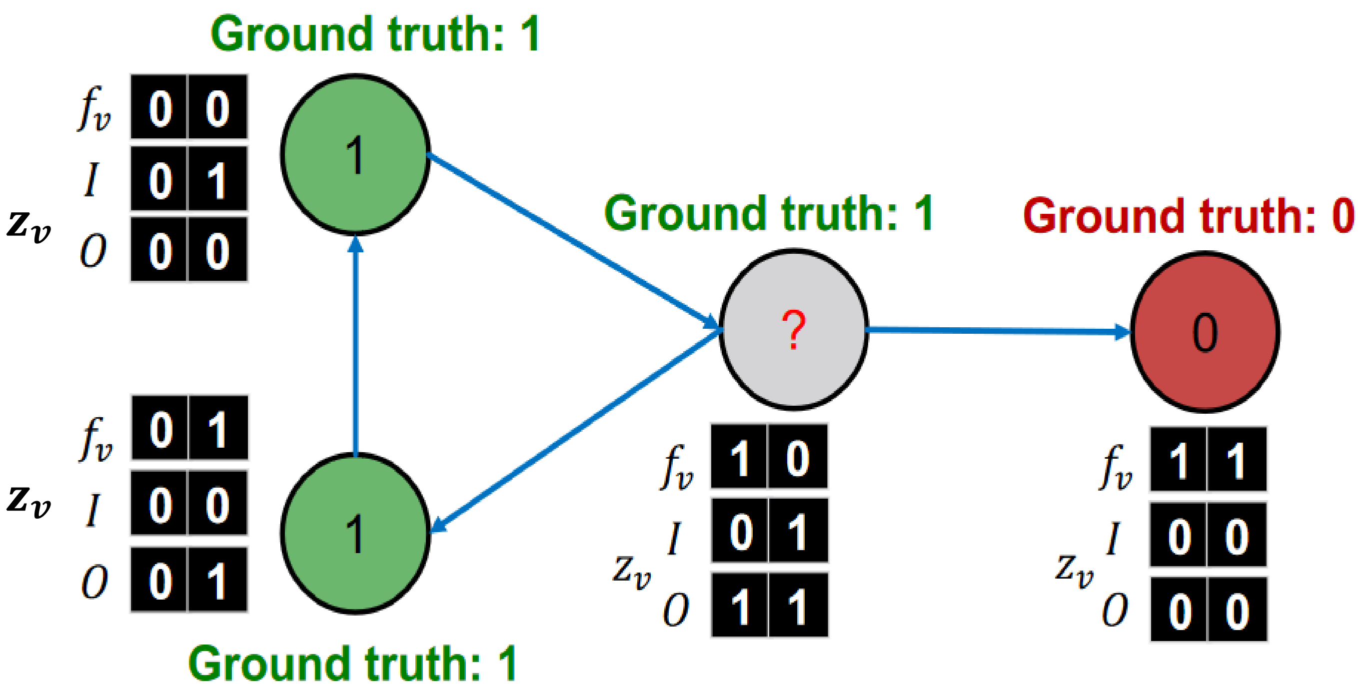
# Example: Web Page Classification

- Baseline: train a classifier (e.g., linear classifier) to classify pages based on binary node attributes.



# Example: Web Page Classification

- Each node maintains vectors $\boldsymbol{z_v}$ of neighborhood labels:
  - $I$ = **Incoming** neighbor label information vector.
  - $O$= **Outgoing** neighbor label information vector.
    - $I_0 = 1$ if at least one of the incoming pages is labelled 0. Similar definitions for $I_1$, $O_0$, and $O_1$

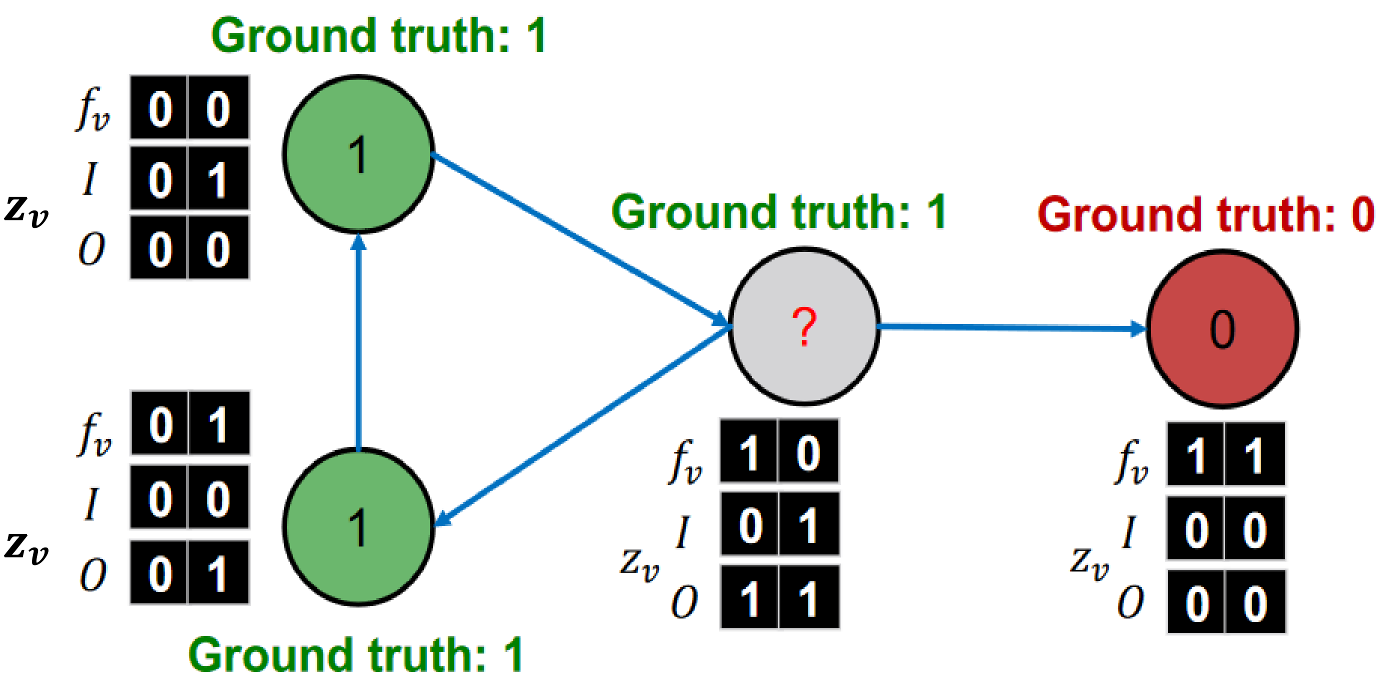



Include network features

# Iterative Classifier – Step 1

- On **training labels**, train two classifiers:
  - Node attribute vector only: $\phi_1(f_v)$
  - Node attribute and link vectors $z_v$: $\phi_2(f_v, z_v)$

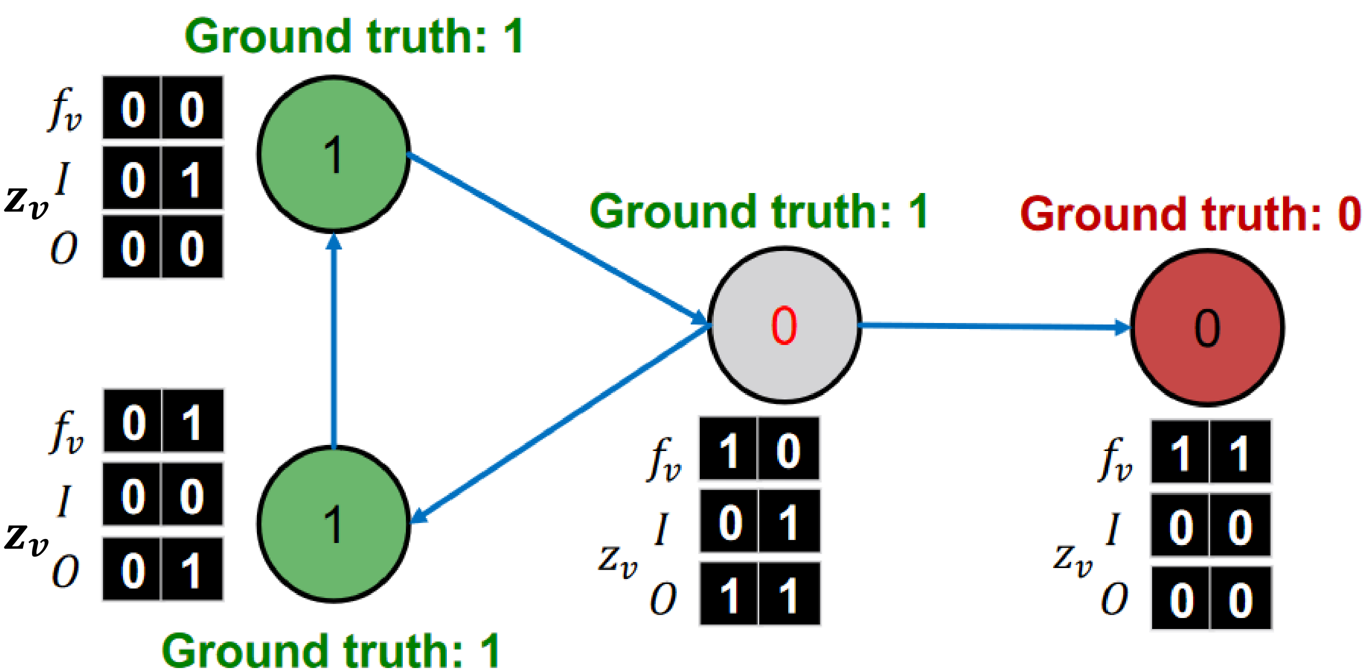
**1. Train classifiers**
**2. Apply classifier to unlab. set**
**3. Iterate**
**4. Update relational features $z_v$**
**5. Update label $Y_v$**



# Iterative Classifier – Step 2

- On the **unlabeled set**:
  - Use trained node feature vector classifier $\phi_1$ to set $Y_v$

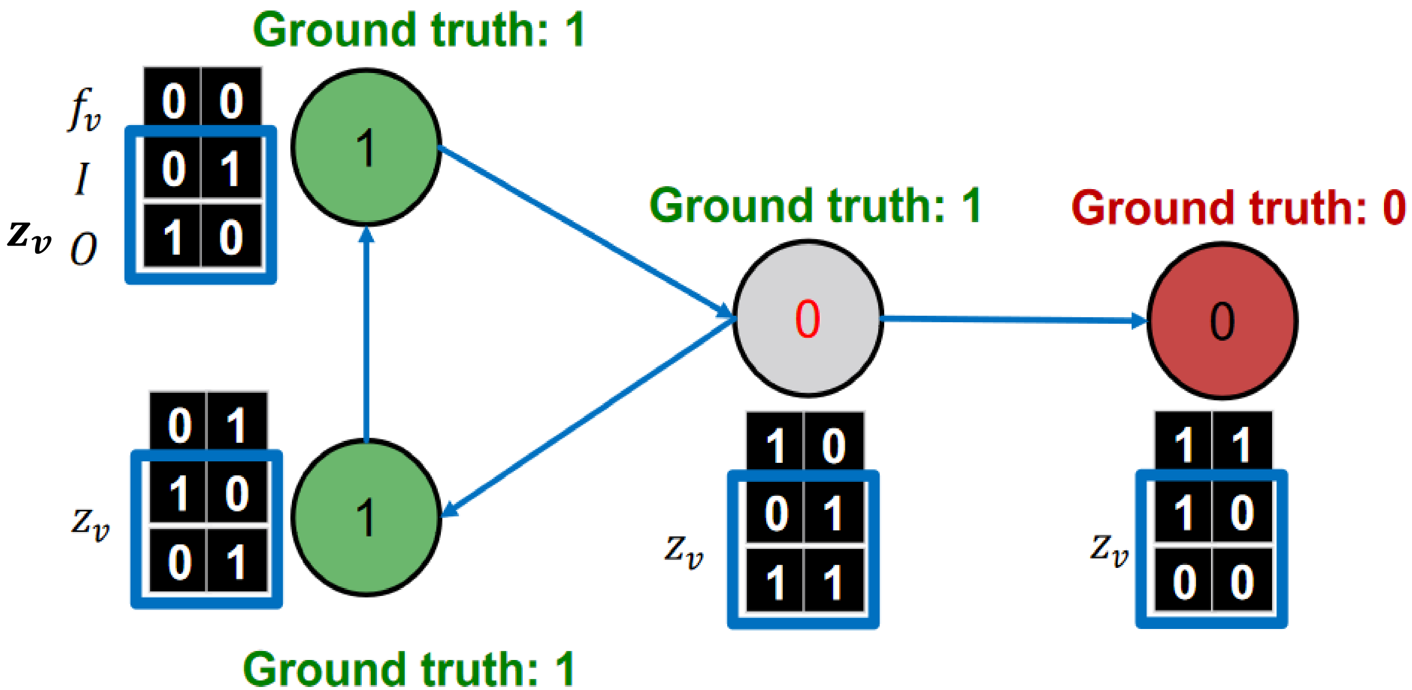
1. Train classifiers
2. **Apply classifier to unlab. set**
3. Iterate
   4. Update relational features $z_v$
   5. Update label $Y_v$



# Iterative Classifier – Step 4

- **Update $z_v$ for all nodes:**

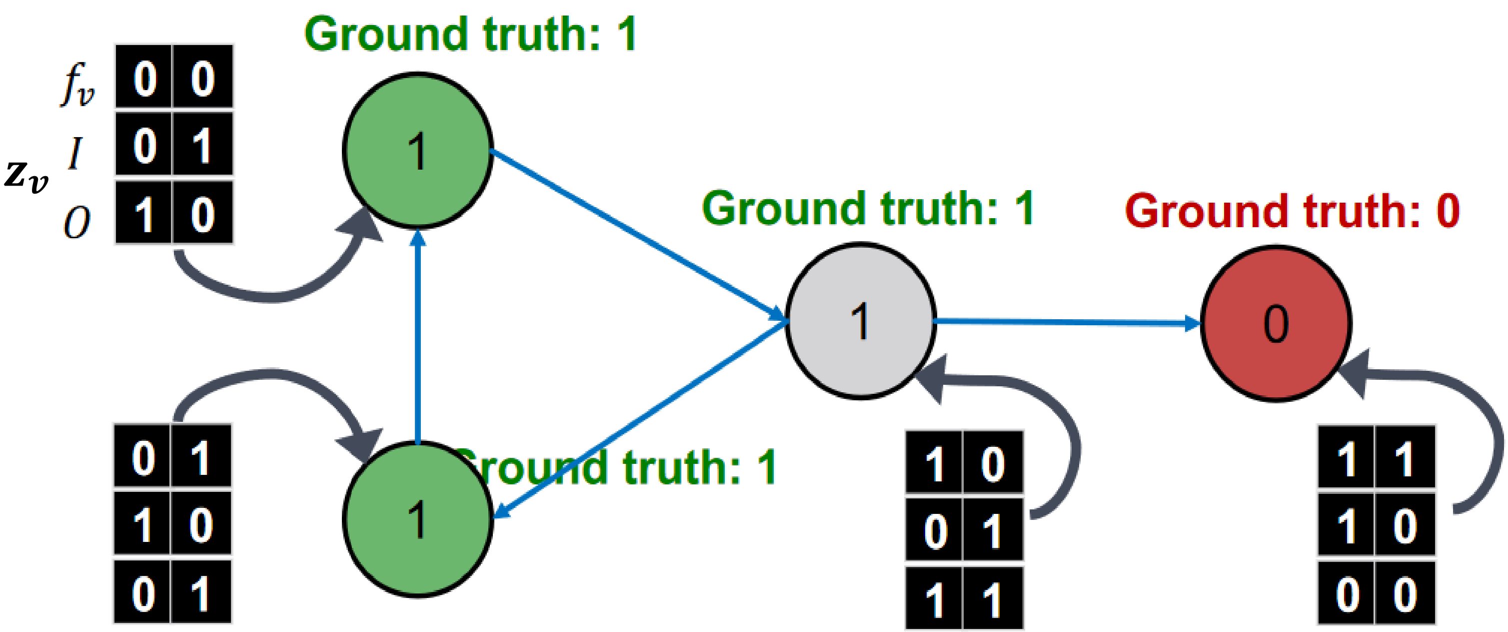
1. Train classifiers
2. Apply classifier to unlab. set
3. Iterate
   4. Update relational features $z_v$
   5. Update label $Y_v$



# Iterative Classifier – Step 5

- **Re-classify all nodes with $\phi_2$:**

1. Train classifiers
2. Apply classifier to unlab. set
3. Iterate
   4. Update relational features $z_v$
   5. Update label $Y_v$

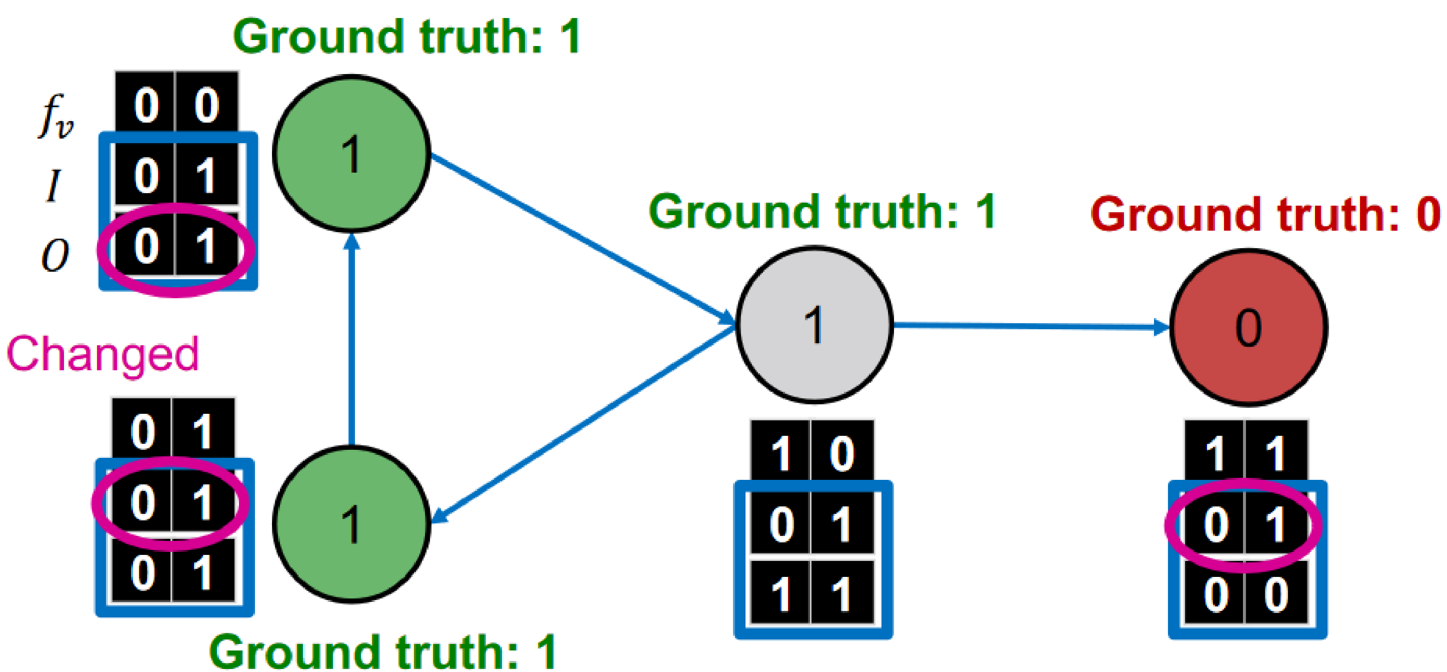



Now it's correct prediction!

# Iterative Classifier - Iterate

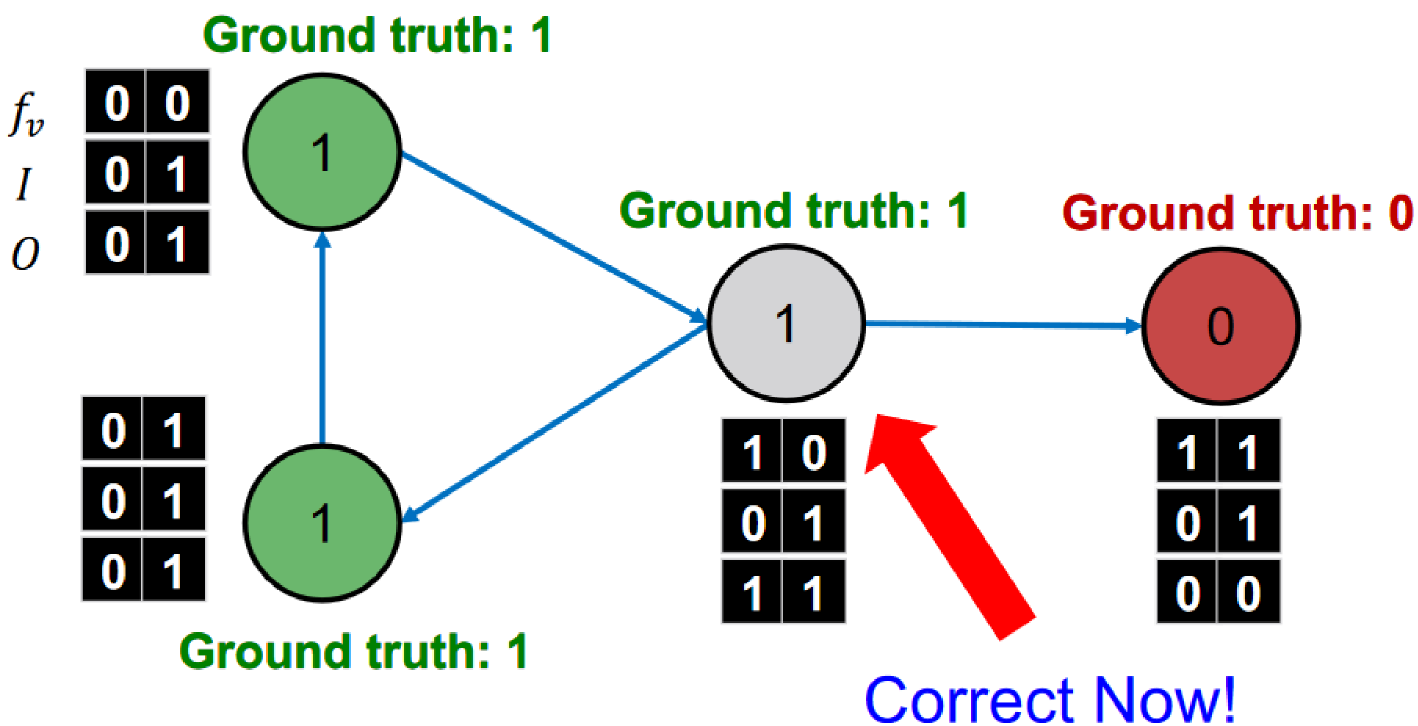
- **Continue until convergence**
  - Update $z_v$ based on $Y_v$
  - Update $Y_v = \phi_2(f_v, z_v)$

1. Train classifiers
2. Apply classifier to unlab. set
3. Iterate
   4. Update relational features $z_v$
   5. Update label $Y_v$



# Iterative Classifier – Prediction

- **Stop iteration**
  - After convergence or when maximum iterations are reached



# 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 $i$ based on its **features** as well as **labels** of neighbors