# Classification

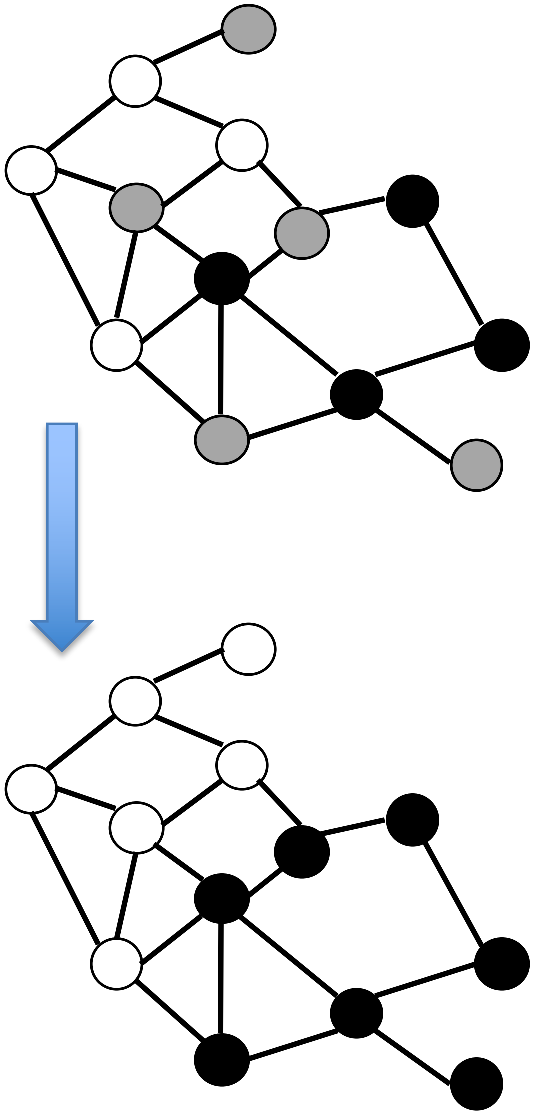
Social and Technological Networks

Rik Sarkar

University of Edinburgh, 2019.

# Assign labels to unlabeled vertices

- Vertices are labeled
  - White: 0
  - Black: 1
  - Grey: Unknown
- Determine labels for grey vertices

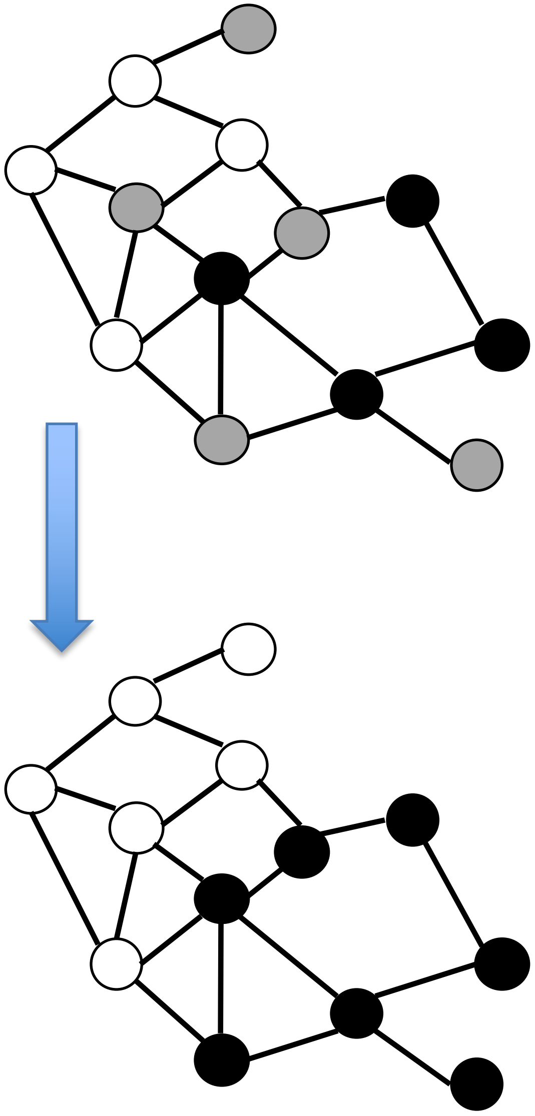


# Assign labels to unlabeled vertices

- Semi-supervised learning

- Example query points are known (vertices)

- But their labels are not known



# Attribute labelling is possible

- Because of homophily

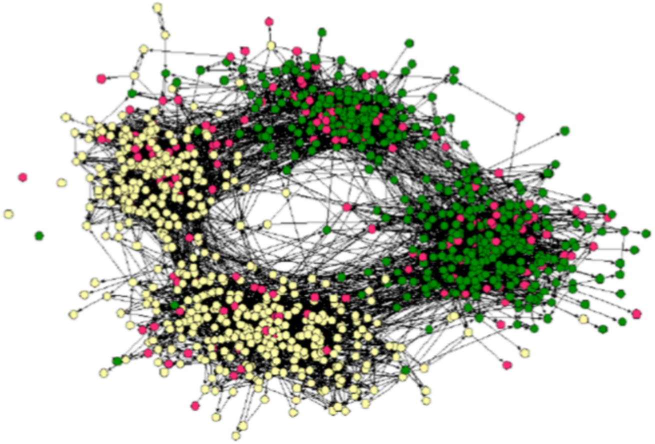
- Nodes with similar attributes are more likely to be connected, and vice versa

- Nodes in the same community are more likely to have similar labels

- Labels depend on other labels, labels of neighbors

# Examples

- People of the same race are more often connected (see Easley-kleinberg)
- Malicious web sites link to other malicious web sites
- Sites on a topic link to other sites on the topic



# Other applications

- Document classification
- Tagging
- Link pregiction
- Image/data segmentation
- Spam and fraud detection
- …

# Iterative weighted averaging

- Weights w of edges
- Labels L of nodes
  - Labeled node x: L(x) = Label(x)
  - Unlabeled nodes u: L(u) = 0.1 (or any fixed value)
- For *Unlabeled* node u with neighbors N(u)
  - Set $$L(u) = \frac{1}{\sum_{v \in N(u)} w(u,v)} \sum_{v \in N(u)} w(u,v) L(v)$$
  - Repeat
- Each node gets a label in range [0, 1]

- To get binary labels

- Set a threshold, e.g. 0.5.

- And
  - If $L(u) \leq 0.5$, set $\text{Label}(u) = 0$
  - If $L(u) > 0.5$, set $\text{Label}(u) = 1$

- Note that prior labeled nodes stay as they were

# Special case

- Unweighted graph
- All weights = 1

- L(u) is just average of neighbor labels
  - Including itself, if defined that way

- Called Harmonic functions
- Convergence guaranteed

# Multiple attributes

- Suppose there are multiple attributes/labels
- There may be correlations between labels
- How do we make use of these attributes?

# Multiple attributes

- Learn a local classifier function f
  - E.g. SVM, kNN, ….
  - Takes a vector A of node labels and outputs a suggested class
- Repeat for each node with unknown label
  - Compute a vector A of neighbor labels
  - With aggregate values of neighbor labels
    - E.g. with mean values, median, etc…
  - Repeat until convergence or MAX iterations
  - (since convergence is not guaranteed)