CSE8803/CX4803

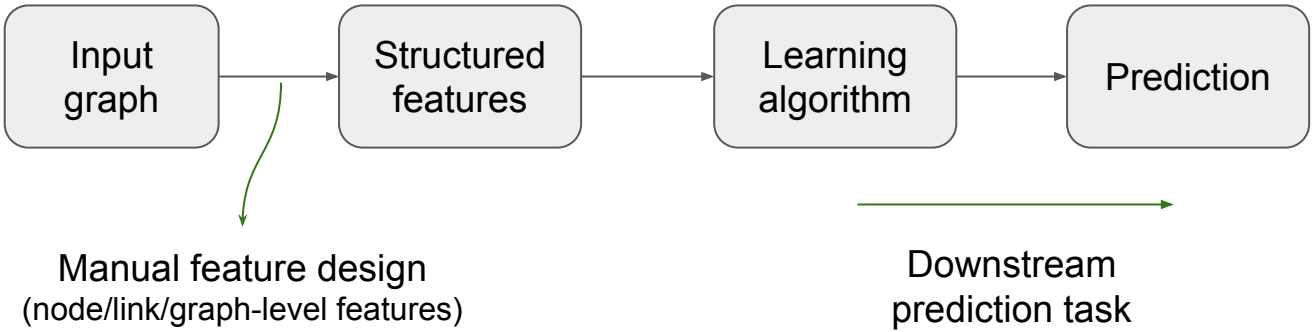
Machine Learning in Computational Biology

Lecture 15:
Representation Learning in Graphs (Network Embeddings)

Yunan Luo

Last lecture: traditional ML for graphs

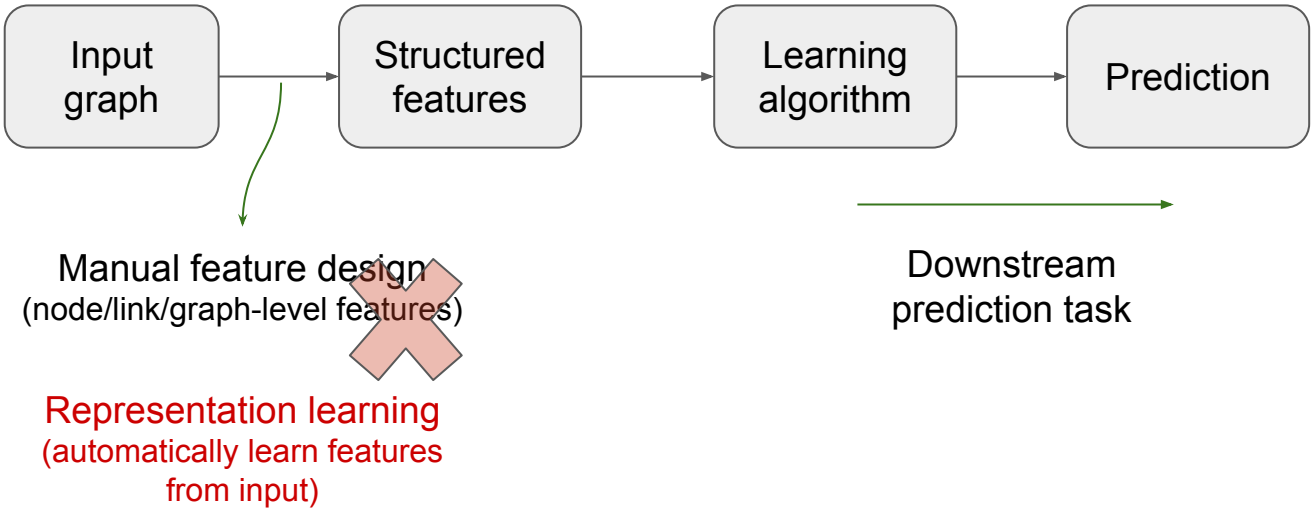
- Given an input graph, extract node, link and graph-level features, learn a model (SVM, neural network, etc.) that maps features to labels.



Adapted from CS224W, Jure Leskovec

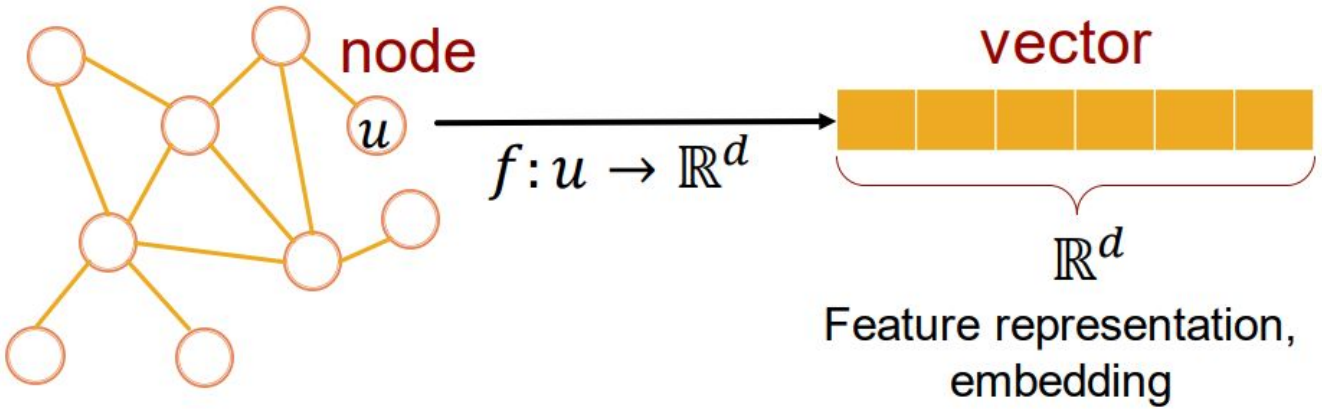
Graph representation learning

- Graph representation learning: learn a feature for each node from the graph input automatically



Graph representation learning

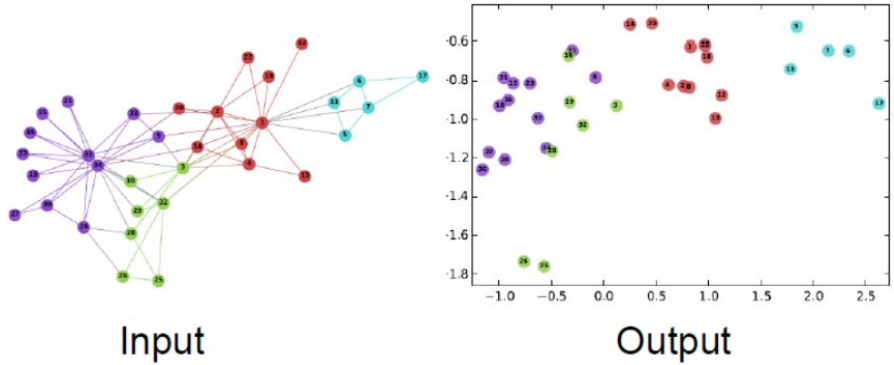
- **Goal**: Efficient task-independent feature learning for machine learning with graphs



Adapted from CS224W, Jure Leskovec

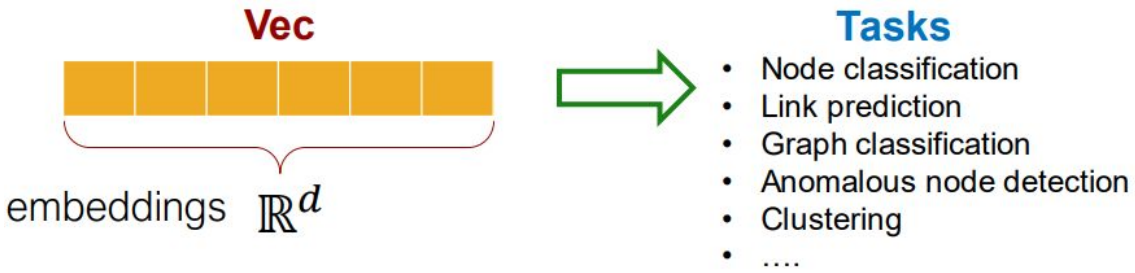
Why embedding?

- Similarity of embeddings between nodes indicates their similarity in the network.



DeepWalk: Online Learning of Social Representations. KDD 2014

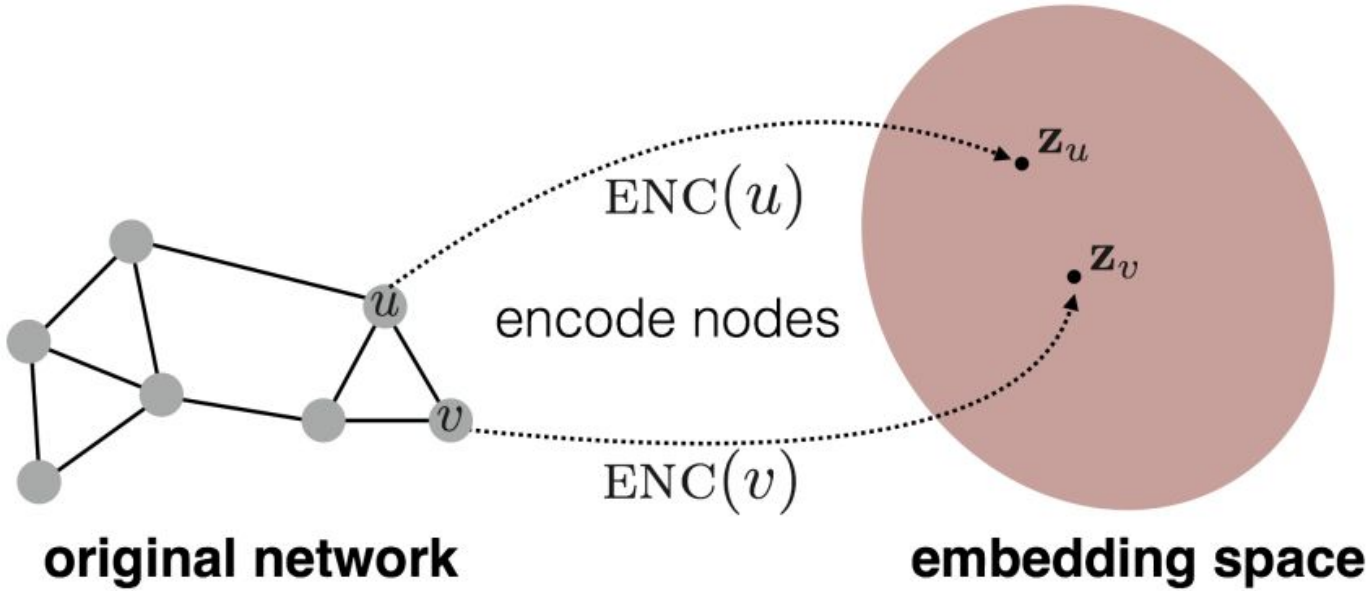
- Potentially used for many downstream predictions



Adapted from CS224W, Jure Leskovec

Embedding nodes

- **Goal**: encode nodes so that similarity in the embedding space (e.g., dot product) approximates similarity in the graph



Adapted from CS224W, Jure Leskovec

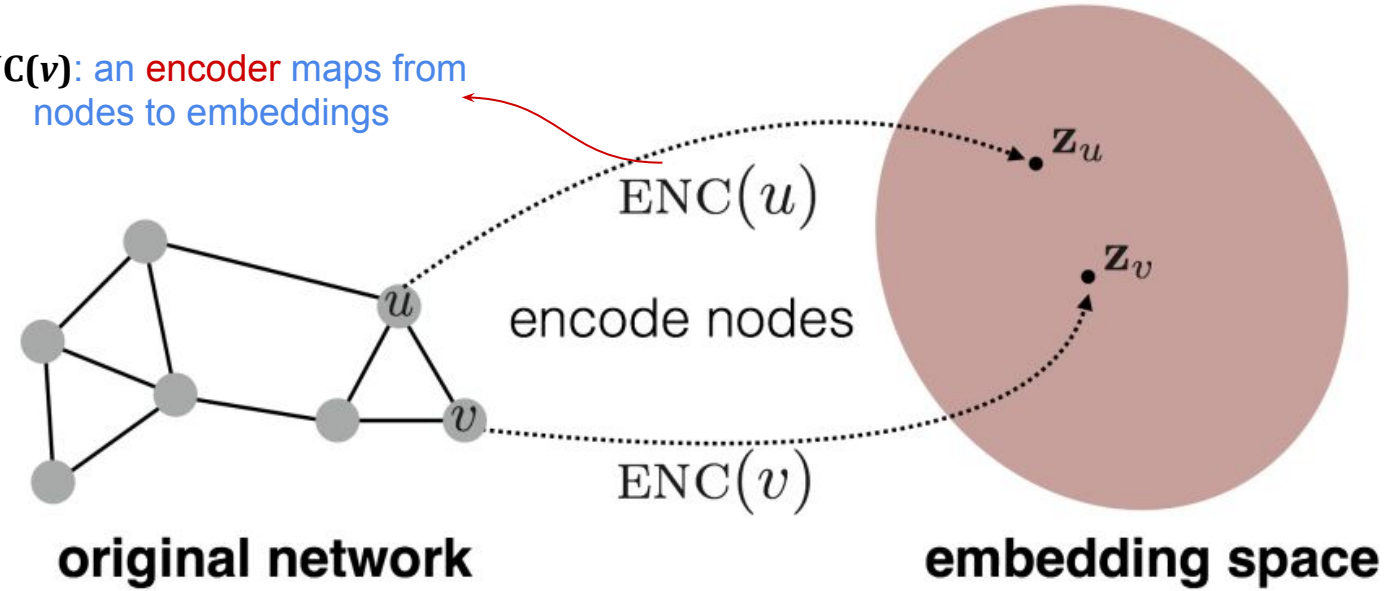
Embedding nodes

$$\text{similarity}(u, v) \approx \mathbf{z}_v^{\mathrm{T}} \mathbf{z}_u$$

In the original network

Similarity of the embeddings

ENC(v): an encoder maps from nodes to embeddings



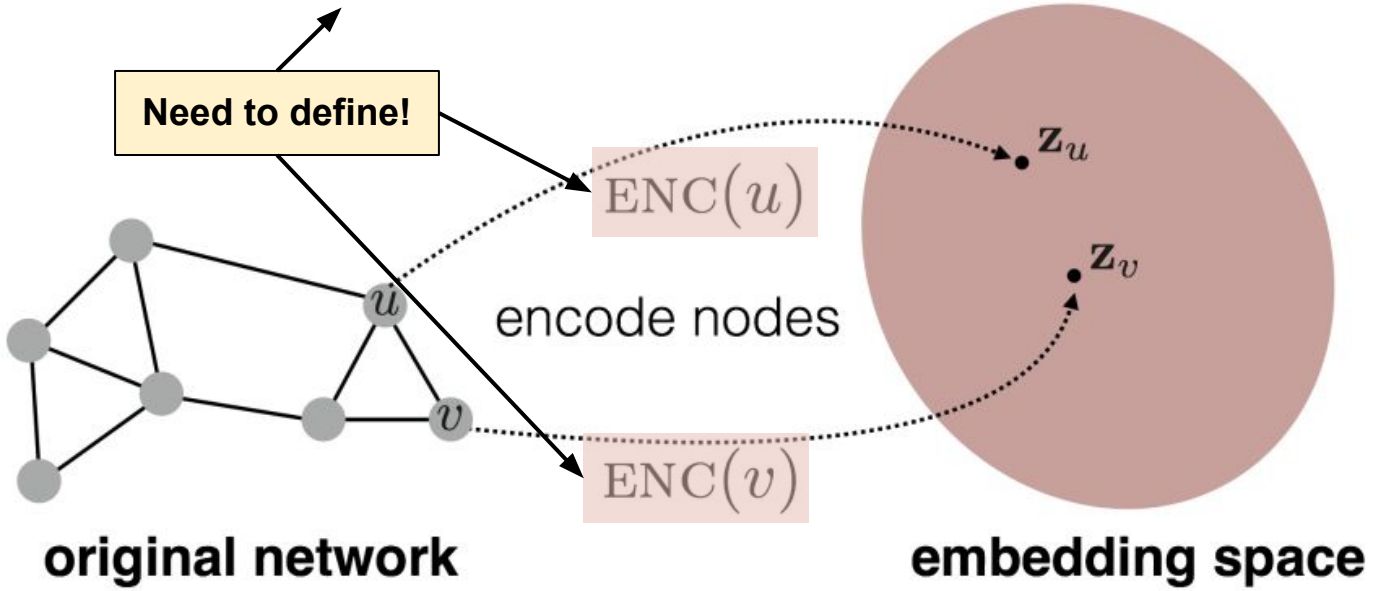
Adapted from CS224W, Jure Leskovec

Embedding nodes

$$\text{similarity}(u, v) \approx \mathbf{z}_v^{\mathrm{T}} \mathbf{z}_u$$

In the original network

Similarity of the embeddings



Adapted from CS224W, Jure Leskovec

Two key components

- **Encoder**: maps each node to a low-dimensional vector

$$\mathrm{ENC}(\boldsymbol{v}) = \mathbf{Z}_v$$

$\mathbf{Z}_v$: *d*-dimensional vector (embedding)

$\boldsymbol{v}$: Node in the input graph

- **Similarity function**: specifies how the relationships in vector space map to the relationships in the original network

$$\text{similarity}(u, v) \approx \mathbf{Z}_v{}^{\mathrm{T}}\mathbf{Z}_u$$

$\text{similarity}(u, v)$: Similarity of $\boldsymbol{u}$ and $\boldsymbol{v}$ in the original network

$\mathbf{Z}_v{}^{\mathrm{T}}\mathbf{Z}_u$: Dot product between node embeddings

Adapted from CS224W, Jure Leskovec

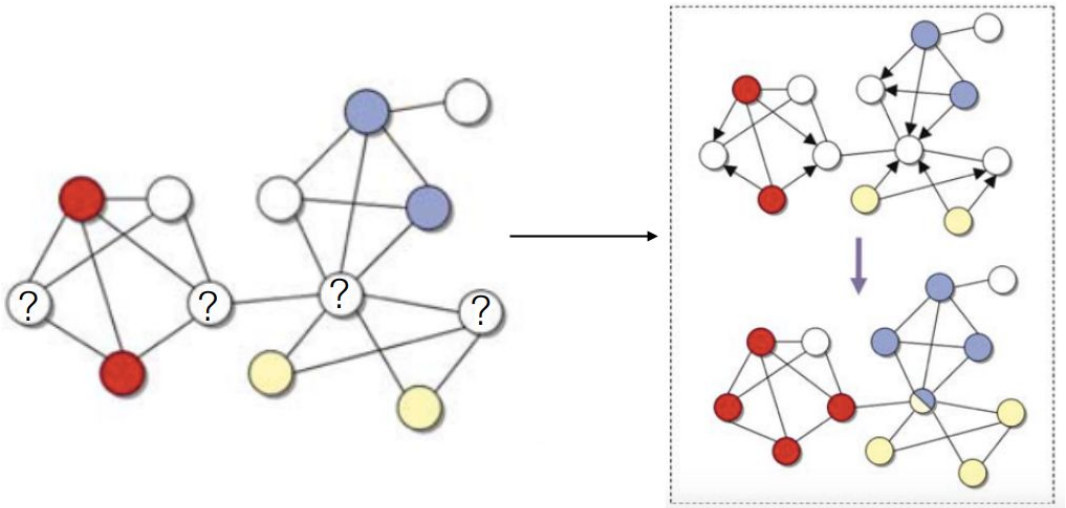
How to define node similarity in the network?

- Key choice of methods is how to define node similarity.
- Should two nodes have a similar embedding if they …
  - are linked?
  - share neighbors?
  - …
- This lecture: define node similarity based "topological roles" of each node with respect to other nodes.
- Two graph representation learning algorithms:
  - Diffusion component analysis [DCA] (Cho et al, 2016, *Cell Systems*)
  - Node2vec (Grover et al, 2016, *KDD*)

Adapted from CS224W, Jure Leskovec

Motivating example

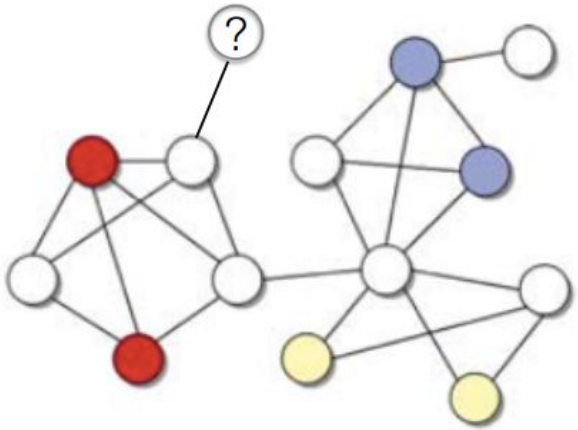
Example: protein function prediction



Voting by direct neighbors

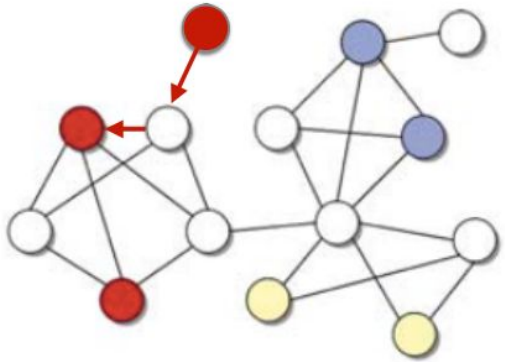
Adapted from CS598, Jian Peng

If there is no direct neighbor with known function



Adapted from CS598, Jian Peng

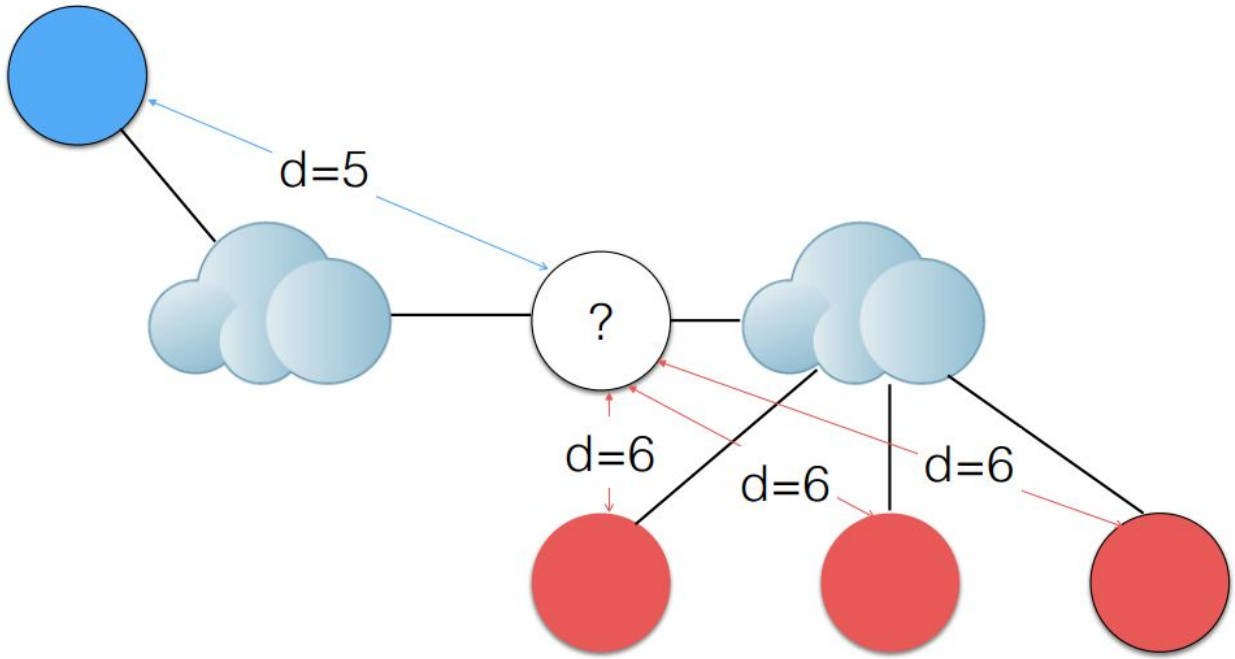
Shortest path



Floyd-Warshall algorithm: all pairwise distances
Computational Complexity: $O(n^3)$

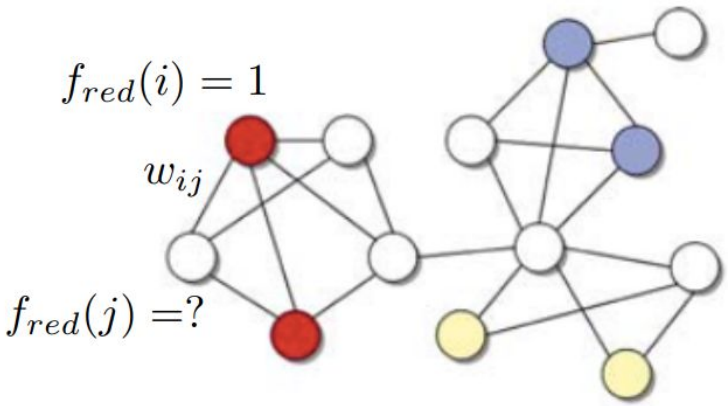
Adapted from CS598, Jian Peng

Is shortest path a good metric?



Adapted from CS598, Jian Peng

Label propagation algorithm



How to solve this problem?

Connected nodes tend to have similar function (color).

$$\min_{f_{red}} \sum_{(i,j)\in E} w_{ij}(f_{red}(i) - f_{red}(j))^2$$

$$\forall i \in RED, f_{red}(i) = 1$$

Adapted from CS598, Jian Peng