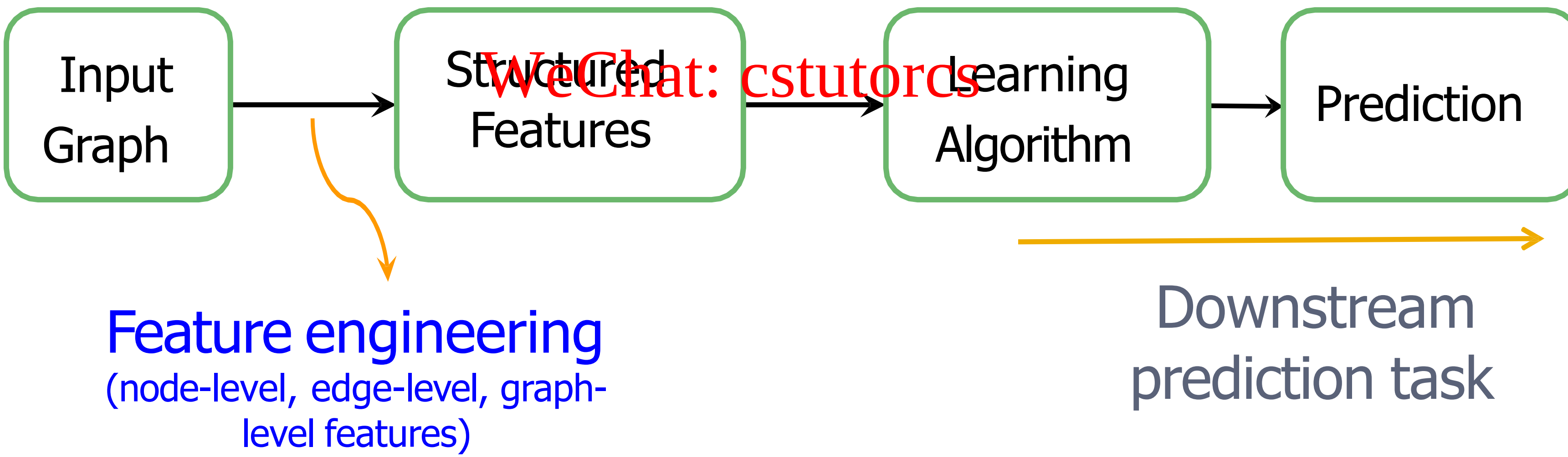
Node Embeddings

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Recap: 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.

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs



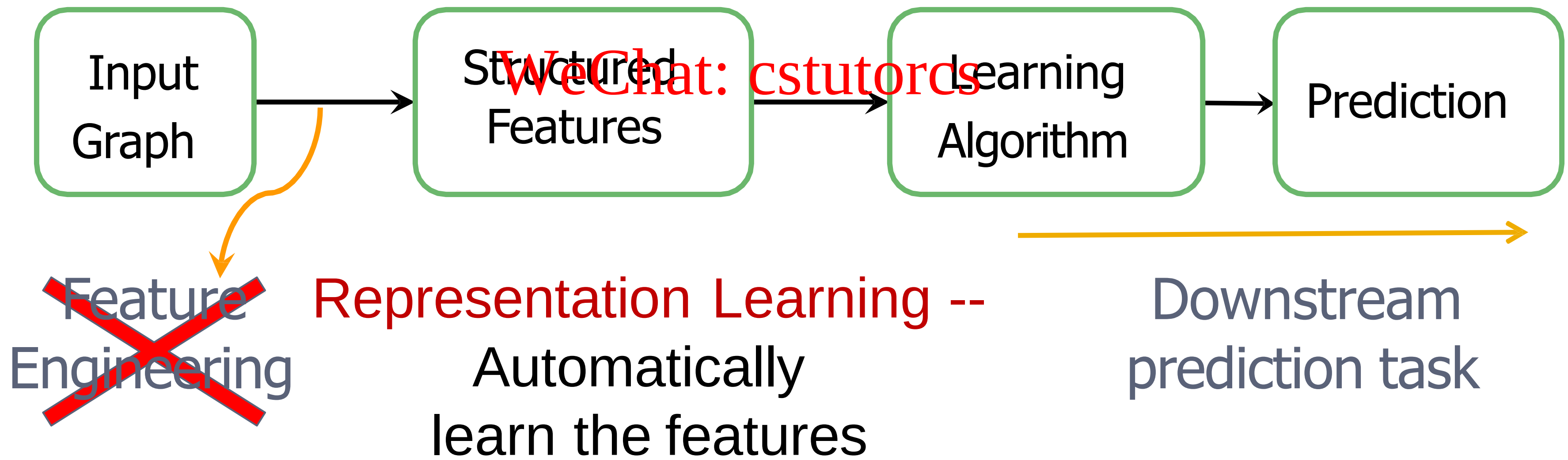
Graph Representation Learning

Graph Representation Learning alleviates the need to do feature engineering every single time.

Assignment Project Exam Help

https://tutorcs.com

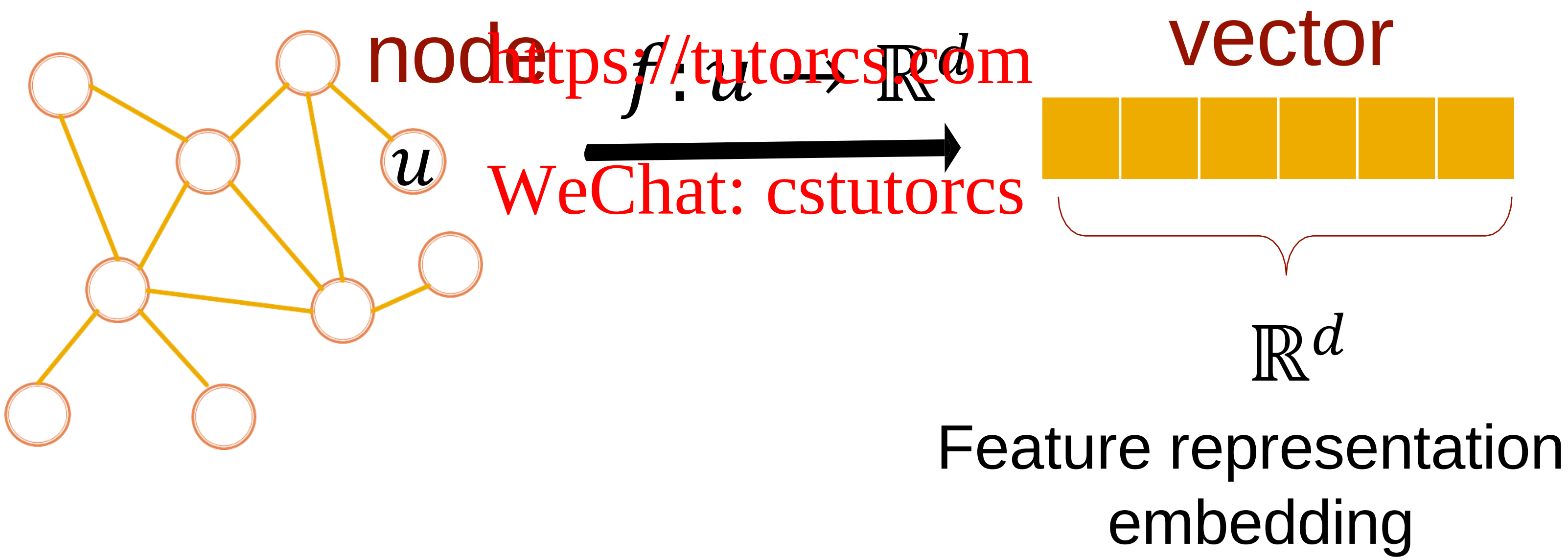
WeChat: cstutorcs



Graph Representation Learning

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

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

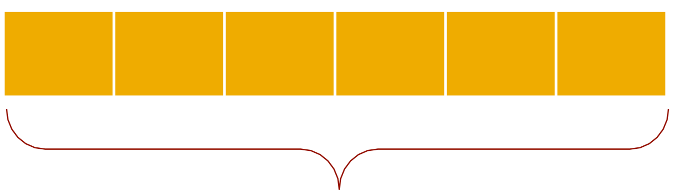


Why Embedding?

- **Task: Map nodes into an embedding space**
  - Similarity of embeddings between nodes indicates their similarity in the network.
    - For example: Both nodes are close to each other (connected by an edge)
  - Encode network information
  - Potentially used for many downstream predictions

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Vec



$\mathbb{R}^d$ embeddings

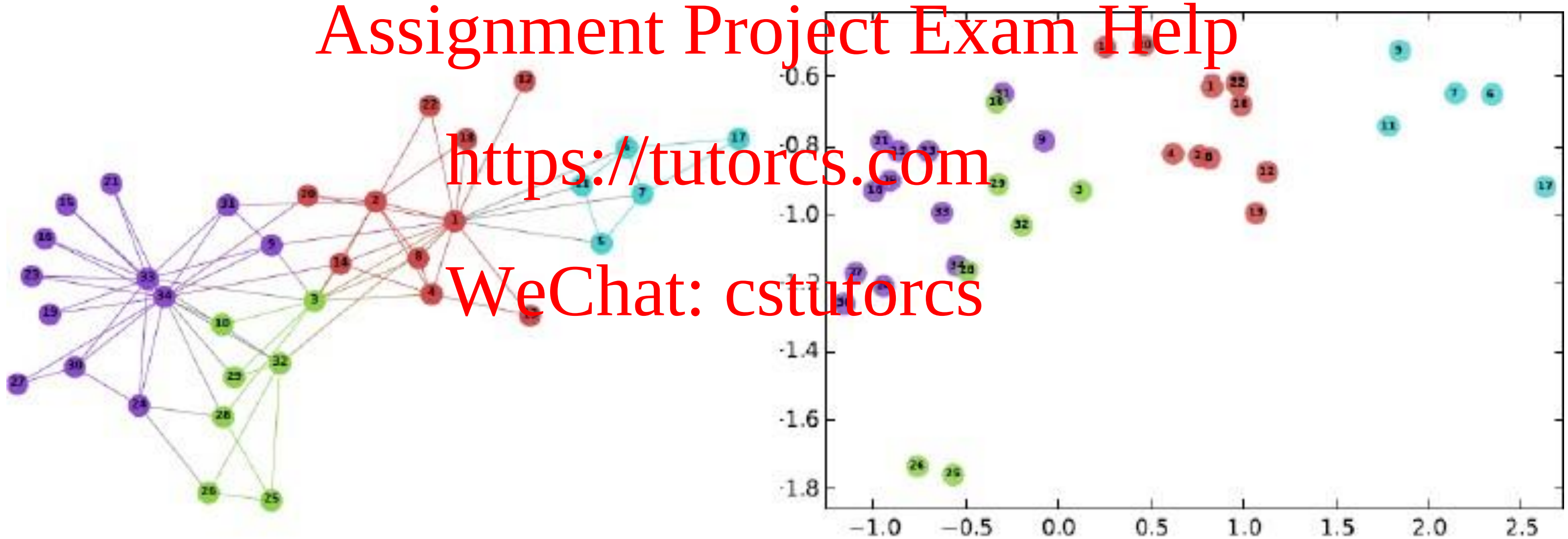
Tasks

- Node classification
- Link prediction
- Graph classification
- Anomalous node detection
- Clustering
- ….

Example Node Embedding

- 2D embedding of nodes of the Zachary's Karate Club network:

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs



Input Output

Image from: [Perozzi et al](). DeepWalk: Online Learning of Social Representations. *KDD 2014.*

Node Embeddings: Encoder and Decoder

Assignment Project Exam Help

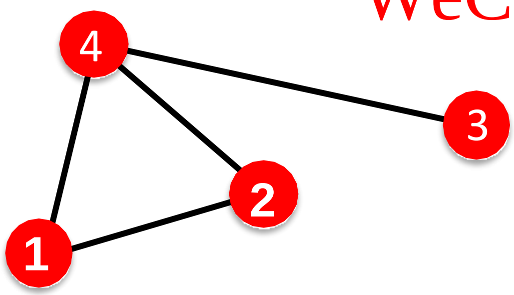
https://tutorcs.com

WeChat: cstutorcs

Setup

- Assume we have a graph G:
  - V is the vertex set.
  - A is the adjacency matrix (assume binary).
  - For simplicity: No node features or extra information is used

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs



V: {1, 2, 3, 4}

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

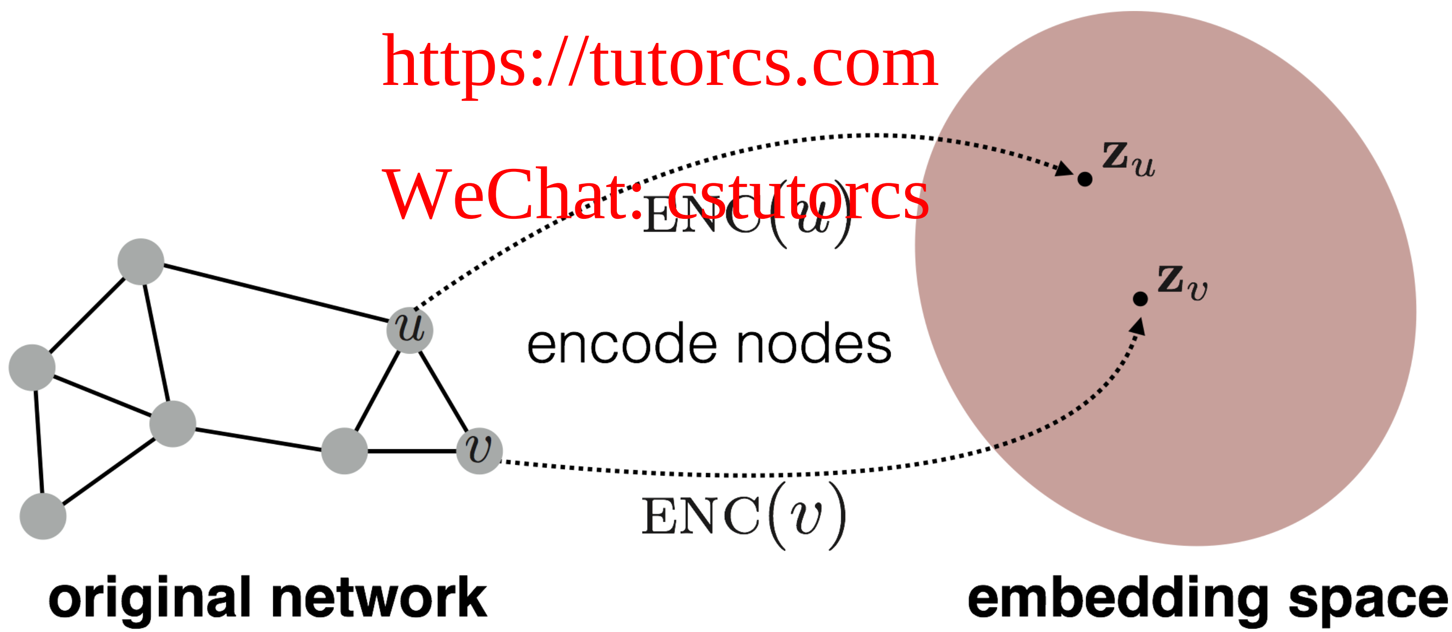
Embedding Nodes

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

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

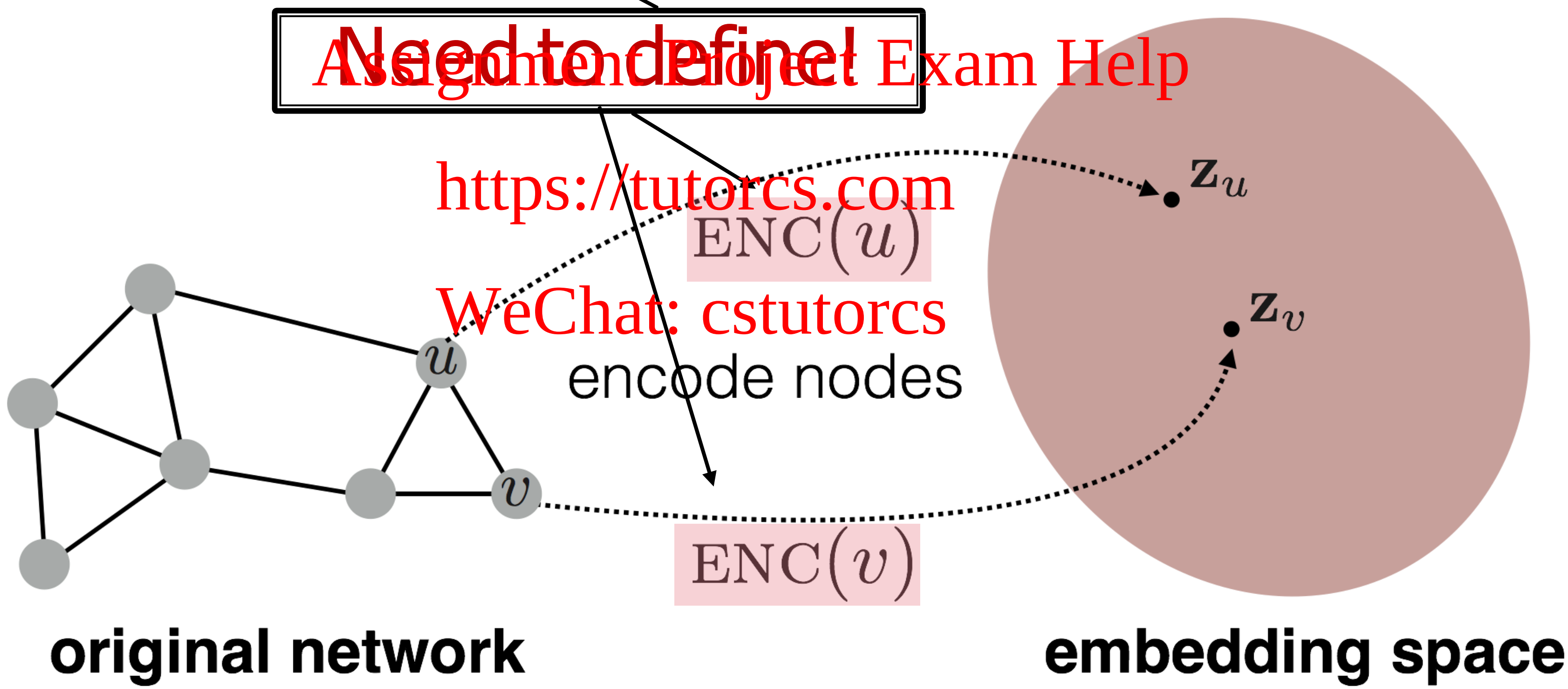


Embedding Nodes

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

in the original network

Similarity of the embedding



Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Node Embeddings Summary

1. Encoder **ENC** maps from nodes to embeddings
2. Define a node similarity function (i.e., a measure of similarity in the original network)
Assignment Project Exam Help
3. Decoder **DEC** maps from embeddings to the similarity score
https://tutorcs.com
WeChat: cstutorcs
4. Optimize the parameters of the encoder so that:

$\mathbf{DEC}(\mathbf{z}_v^{\mathrm{T}}\mathbf{z}_u)$

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

in the original network

Similarity of the embedding

Two Key Components

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

$$\text{ENC}(v) = \boxed{\mathbf{z}_v}$$

d-dimensional embedding

node in the input graph

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutors

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

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

Decoder

Similarity of u and v in the original network

dot product between node embeddings

“Shallow” Encoding

Simplest encoding approach:
Encoder is just an embedding-lookup

Assignment Project Exam Help

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

https://tutorcs.com

$$\mathbf{Z} \in \mathbb{R}^{d \times |\mathcal{V}|}$$

Matrix, each column is a node embedding [what we learn / optimize]

WeChat: cstutorcs

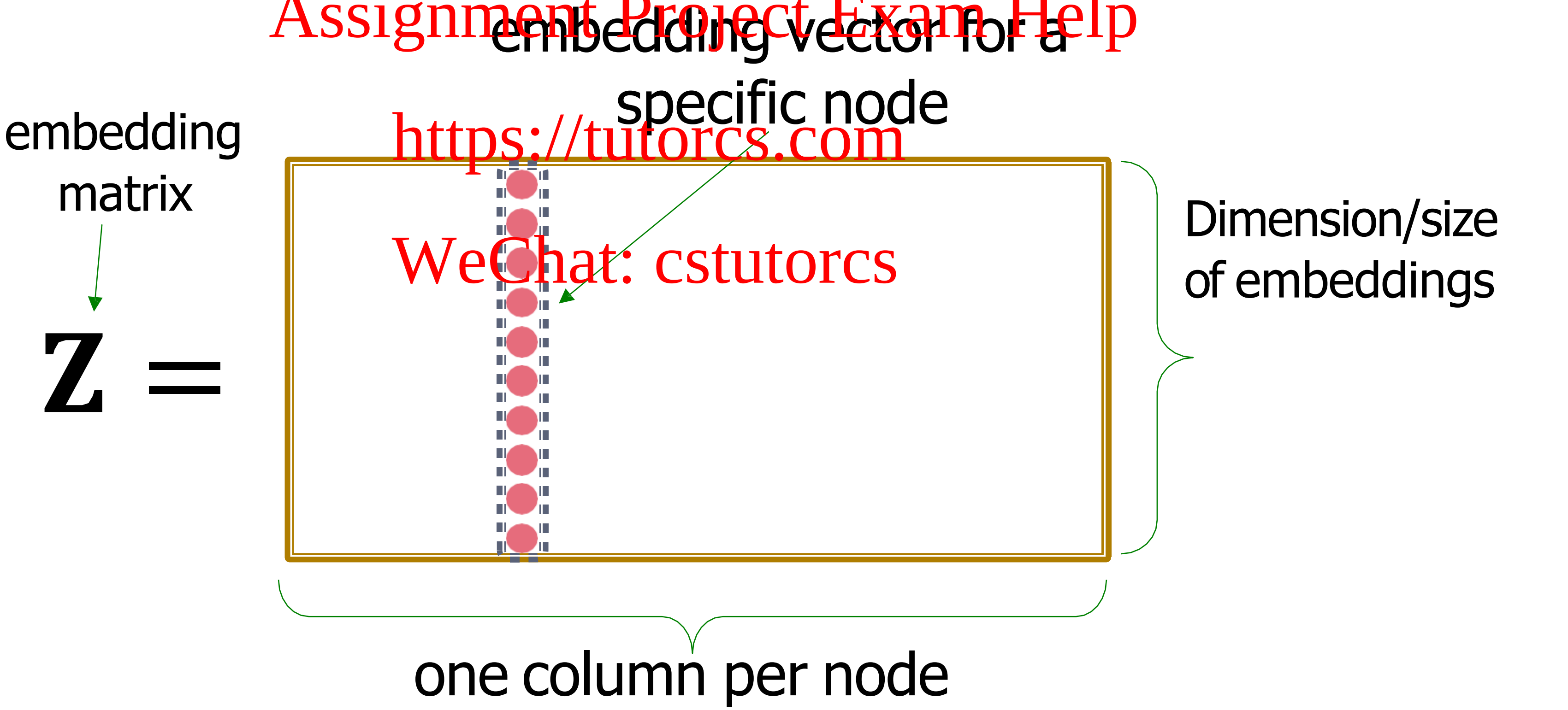
$$v \in \mathbb{I}^{|\mathcal{V}|}$$

Indicator vector, all zeroes except a one in column indicating node *v*

“Shallow” Encoding

Simplest encoding approach:

Encoder is just an embedding-lookup



Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

"Shallow" Encoding

Simplest encoding approach:

Encoder is just an embedding-lookup

Assignment Project Exam Help

Each node is assigned a unique

https://tutorcs.com

embedding vector

WeChat: cstutorcs

(i.e., we directly optimize the embedding of each node)

Framework Summary

- **Encoder + Decoder Framework**
  - Shallow encoder: embedding lookup
  - Parameters to optimize: $\mathbf{Z}$ which contains node embeddings $\mathbf{z}_u$ for all nodes $u \in V$
  - We will cover deep encoders (GNNs)
  - **Decoder:** based on node similarity.
  - **Objective:** maximize $\mathbf{z}_v^{\mathrm{T}}\mathbf{z}_u$ for node pairs (u, v) that are **similar**

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

How to Define Node Similarity

- Key choice of methods is **how they define node similarity.**
- Should two nodes have a similar embedding if they…
  - are linked?
  - share neighbors?
  - have similar "structural roles"?

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Random Walk Approaches for Node Embeddings

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

A Note on Node Embeddings

- We will now learn node similarity definition that uses **random walks**, and how to optimize embeddings for such a similarity measure.
- Random walks is **unsupervised/self-supervised** way of learning node embeddings.
  - We are **not** utilizing node labels
  - We are **not** utilizing node features
- These embeddings are **task independent**
  - They are not trained for a specific task but can be used for any task.

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Notation

- **Vector** $\mathbf{z}_u$:
  - The embedding of node u (what we aim to find).

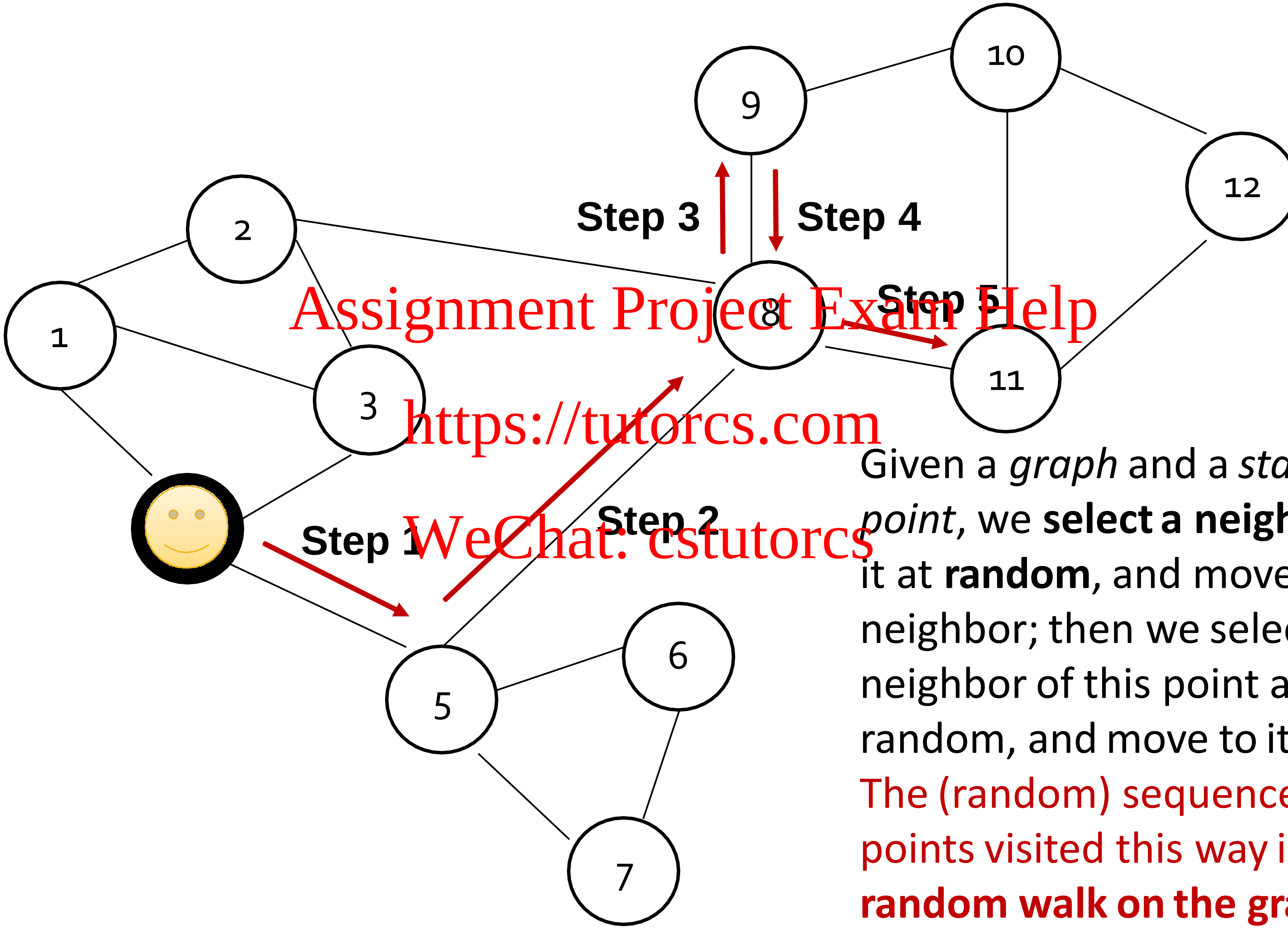
- **Probability** $P(v \mid \mathbf{z}_u)$:
  - The **(predicted) probability** of visiting node v on random walks starting from node u.

Assignment Project Exam Help
https://tutorcs.com

Our model prediction based on $\mathbf{z}_u$

WeChat: cstutorcs

Random Walk



Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Given a *graph* and a *starting point*, we **select a neighbor** of it at **random**, and move to this neighbor; then we select a neighbor of this point at random, and move to it, etc. The (random) sequence of points visited this way is a **random walk on the graph**.

Random Walk Embeddings

Assignment Project Exam Help

https://tutorcs.com

$$\mathbf{z}_u^{\mathrm{T}}\mathbf{z}_v \approx$$ probability that u and v **co-occur** on a random walk over the graph

WeChat: cstutorcs

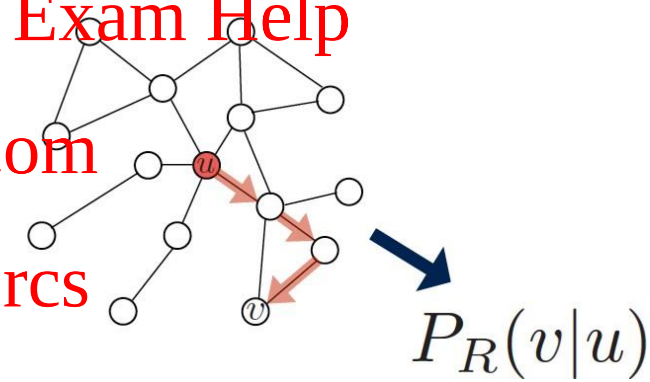
Random Walk Embeddings

1. Estimate probability of visiting node $\boldsymbol{v}$ on a random walk starting from node $\boldsymbol{u}$ using some random walk strategy $\boldsymbol{R}$

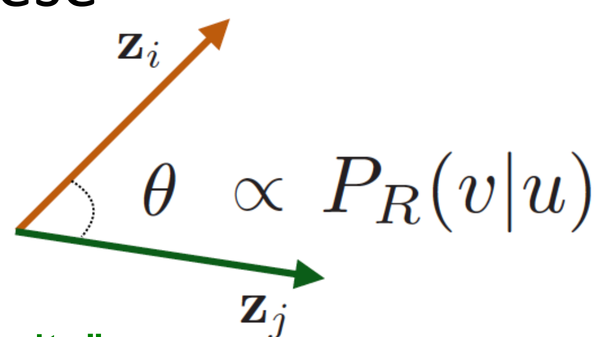
Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs



2. Optimize embeddings to encode these random walk statistics:



Similarity in embedding space (Here:
dot product=$\cos(\theta)$) encodes random walk “similarity”

Why Random Walks

1. **Expressivity:** Flexible stochastic definition of node similarity that incorporates both local and higher-order neighborhood information.
   Idea: if random walk starting from node u visits v with high probability, u and v are similar (high-order multi-hop information).

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

2. **Efficiency:** Do not need to consider all node pairs when training; only need to consider pairs that co-occur on random walks.

Unsupervised Feature Learning

- Intuition: Find embedding of nodes in d-dimensional space that preserves **similarity**.
Assignment Project Exam Help
- Idea: Learn node embedding such that nearby nodes are close together in the network.
https://tutorcs.com
WeChat: cstutorcs
- Given a node u, how do we define nearby nodes?
  - $N_R(u)$ … neighbourhood of u obtained by some random walk strategy R.

Feature Learning as Optimization

- Given $G = (V, E)$,
- Our goal is to learn a mapping $f: u \to \mathbb{R}^d$:

  $f(u) = \mathbf{z}_u$

Assignment Project Exam Help

- Log-likelihood objective:

https://tutorcs.com

$$\max_f \sum_{u \in V} \log \mathrm{P}(N_{\mathrm{R}}(u) \mid \mathbf{z}_u)$$

WeChat: cstutorcs

  - $N_R(u)$ is the neighborhood of node u by strategy R

- Given node u, we want to learn feature representations that are predictive of the nodes in its random walk neighborhood $N_R(u)$.

Random Walk Optimization

1. Run **short fixed-length random walks** starting from each node u in the graph using some random walk strategy R.
2. For each node u collect $N_R(u)$, the multiset* of nodes visited on random walks starting from u.
3. Optimize embeddings according to: Given node u, predict its neighbors $N_{\mathrm{R}}(u)$.

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

$$\max_{f} \sum_{u \in V} \log \mathrm{P}(N_{\mathrm{R}}(u) | \mathbf{z}_u)$$

Maximum likelihood objective

*$N_R(u)$ can have repeat elements since nodes can be visited multiple times on random walks

Random Walks: Summary

1. Run **short fixed-length** random walks starting from each node on the graph
2. For each node u collect $N_R(u)$, the multiset of nodes visited on random walks starting from u.
3. Optimize embeddings (using Stochastic Gradient Descent):

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} -\log(P(v|\mathbf{z}_u))$$

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

How should we Randomly walk?

- So far we have described how to optimize embeddings given a random walk strategy R
- **What strategies should we use to run these random walks?**
  - Simplest idea: **Just run fixed-length, unbiased random walks starting from each node**
  - (i.e., DeepWalk from Perozzi et al., 2013)
    - The issue is that such notion of similarity is too constrained
- **How can we generalize this?**

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Reference: Perozzi et al. 2014. DeepWalk: Online Learning of Social Representations. *KDD.*

Overview of node2vec

- **Goal:** Embed nodes with similar network neighborhoods close in the feature space.
- We frame this goal as a maximum likelihood optimization problem, **independent** to the downstream prediction task.
- **Key observation:** Flexible notion of network neighborhood $N_R(u)$ of node u leads to rich node embeddings
- Develop biased 2nd order random walk R to generate network neighborhood $N_R(u)$ of node u

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Reference: Grover et al. 2016. [node2vec: Scalable Feature Learning for Networks](). *KDD.*

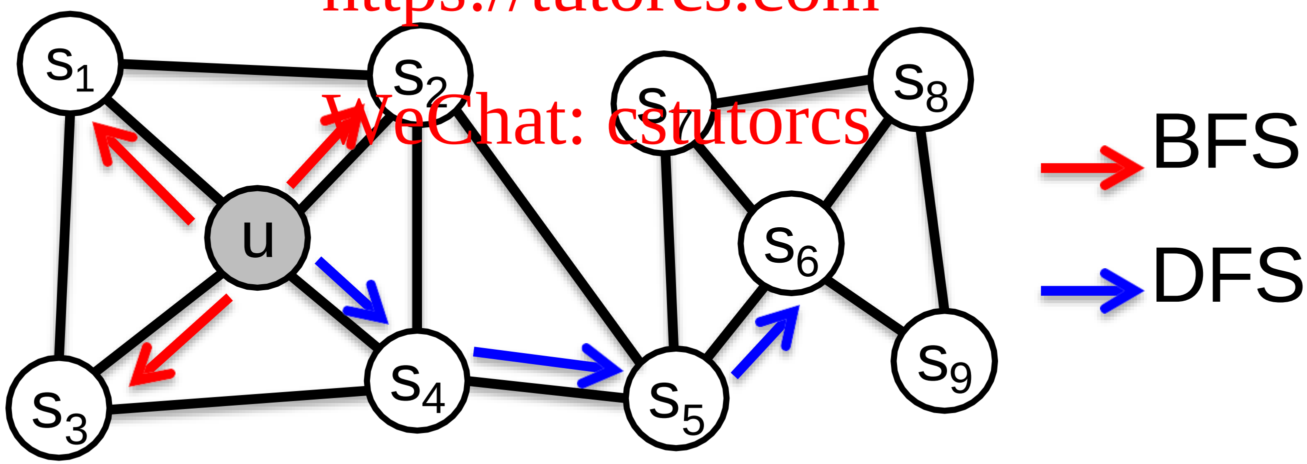
Node2vec: Biased Walks

Idea: use **flexible**, **biased** random walks that can trade off between local and global views of the network (Grover and Leskovec, 2016)

Assignment Project Exam Help

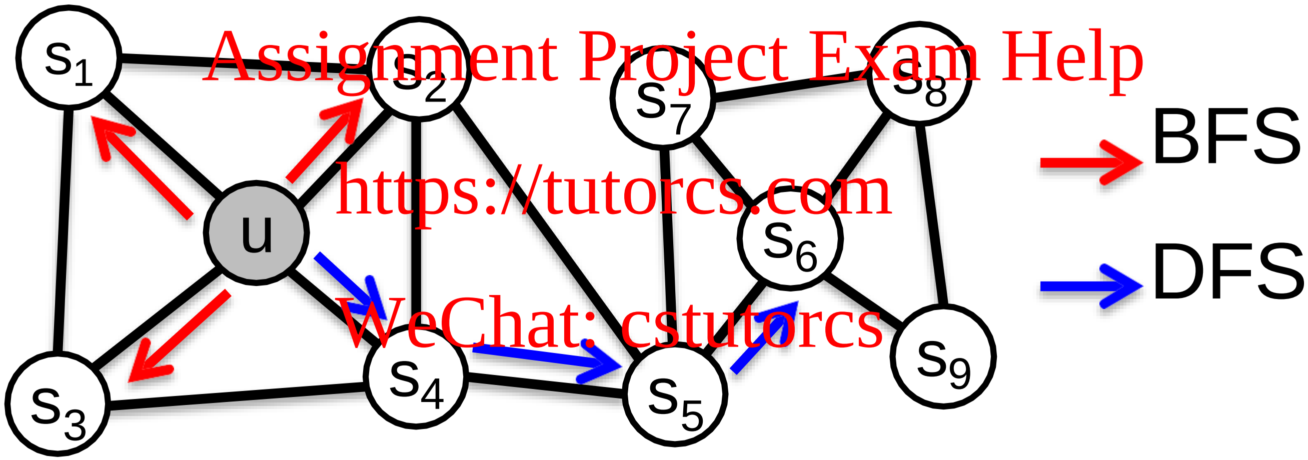
https://tutorcs.com

WeChat: cstutorcs



Node2vec: Biased Walks

Two classic strategies to define a neighborhood $\boldsymbol{N_R(u)}$ of a given node $\boldsymbol{u}$:



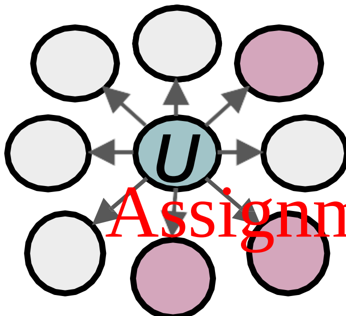
Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Walk of length 3 ($N_R(u)$ of size 3):

$N_{BFS}(u) = \{ s_1, s_2, s_3\}$ Local microscopic view

$N_{DFS}(u) = \{ s_4, s_5, s_6\}$ Global macroscopic view

BFS vs. DFS

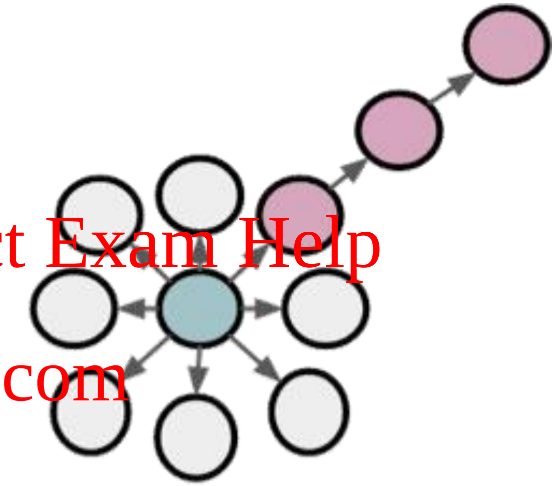


Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

BFS:
Micro-view of neighbourhood



DFS:
Macro-view of neighbourhood

Interpolating BFS and DFS

Biased fixed-length random walk $\boldsymbol{R}$ that given a node $\boldsymbol{u}$ generates neighborhood $\boldsymbol{N_R(u)}$

Assignment Project Exam Help

- Two parameters:
  - **Return parameter $\boldsymbol{p}$:**
    - Return back to the previous node
  - **In-out parameter $\boldsymbol{q}$:**
    - Moving outwards (DFS) vs. inwards (BFS)
    - Intuitively, q is the "ratio" of BFS vs. DFS

https://tutorcs.com

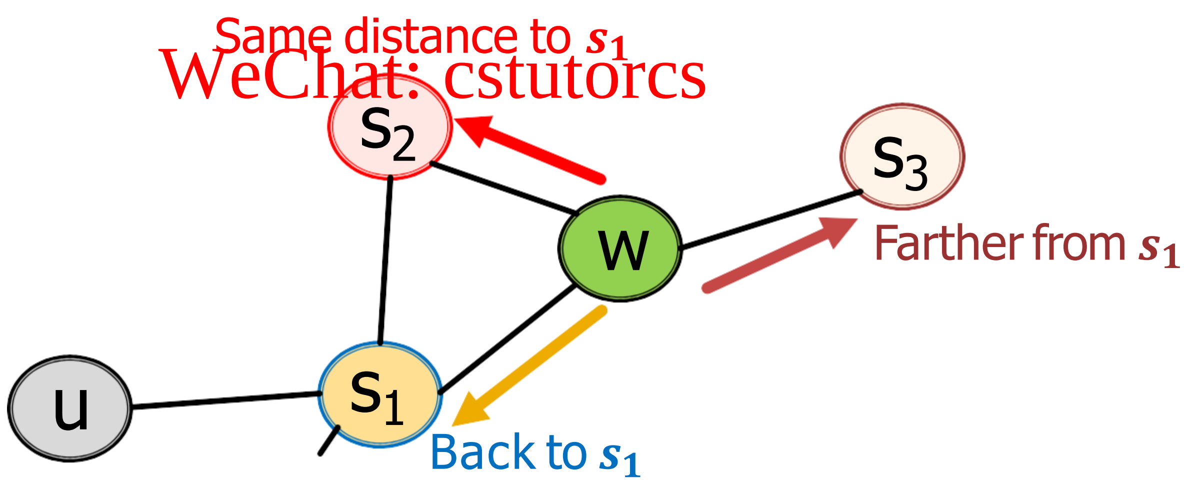
WeChat: cstutorcs

Biased Random Walks

Biased 2nd-order random walks explore network neighborhoods:

- Rnd. walk just traversed edge (s_1, w) and is now at w
- **Insight:** Neighbors of w can only be:

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

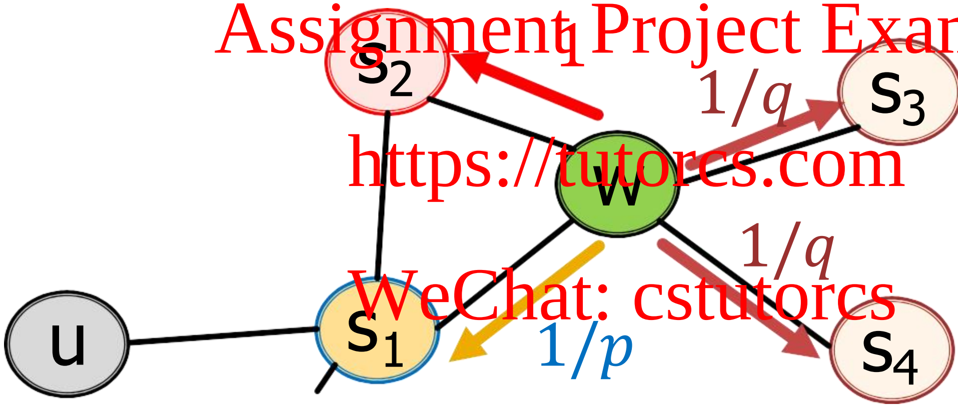


Idea: Remember where the walk came from

Biased Random Walks

- **Walker came over edge (s_1, w) and is at w. Where to go next?**

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs



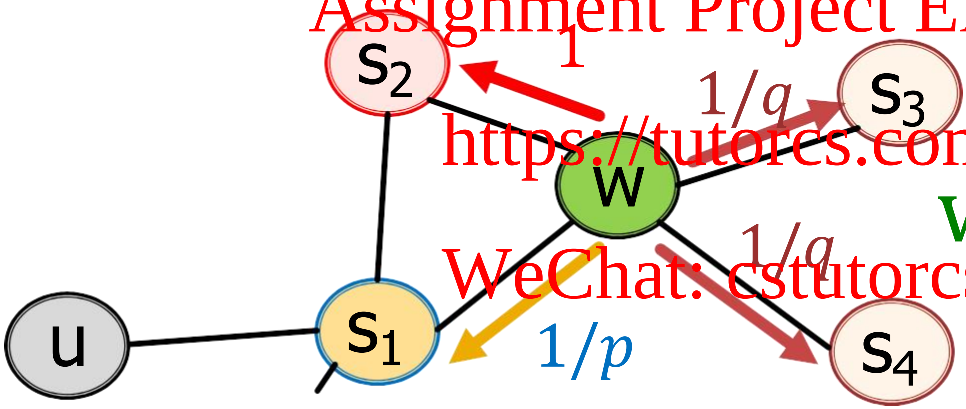
$1/p, 1/q, 1$ are unnormalized probabilities

- p, q model transition probabilities
  - p … return parameter
  - q … ”walk away” parameter

Biased Random Walks

- **Walker came over edge (s_1, w) and is at w. Where to go next?**

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs



$w \rightarrow$

Target t	Prob.	Dist. (s_1, t)
s_1	$1/p$	0
s_2	1	1
s_3	$1/q$	2
s_4	$1/q$	2

Unnormalized transition prob. segmented based on distance from s_1

  - **BFS-like** walk: Low value of p
  - **DFS-like** walk: Low value of q

$N_R(u)$ are the nodes visited by the biased walk

Node2vec Algorithm

1. Compute random walk probabilities
2. Simulate r random walks of **length** l starting from each node u
3. Optimize the node2vec objective (using Stochastic Gradient Descent)

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

Properties:

1) **Linear-time** complexity

2) All 3 steps are individually parallelizable

Summary so far

- **Core idea:** Embed nodes so that distances in embedding space reflect node similarities in the original network.

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

- **Different notions of node similarity:**
  - Naïve: similar if two nodes are connected
  - Neighborhood overlap (covered in Topic 5)
  - Random walk approaches

Summary so far

- No one method wins in all cases....
  - E.g., node2vec performs better on node classification while alternative methods perform better on link prediction ([Goyal and Ferrara, 2017 survey](#)).

Assignment Project Exam Help
https://tutorcs.com
WeChat: cstutorcs

- Random walk approaches are generally more efficient.

So what method should I use..?
Choose definition of node similarity that matches your application.

Learning Outcomes

We discussed **graph representation learning**, a way to learn **node and graph embeddings** for downstream tasks, **without feature engineering**.

Assignment Project Exam Help

- Encoder-decoder framework:
  - **Encoder: embedding lookup**
  - **Decoder: predict score based on embedding to match node similarity**

https://tutorcs.com

WeChat: cstutorcs

- Node similarity measure: (biased) random walk
  - **Examples: Node2Vec**

Acknowledgement: Jure Leskovec, Stanford University