Graph Neural Networks

Lecture 1

Danila Biktimirov
Applied Computer Science
Neapolis University Pafos

7 February 2025

Overview

About the Course. Organizational Information

- 12 lectures and seminars
- 5 homework assignments (3 theoretical, 2 practical)
- Lectures and seminars on Friday at 9:00

About the Course. Grading

- Final Grade $= 0.7 \cdot O_n + 0.3 \cdot O_e$, rounded arithmetically
- $O_n =$ (Total points scored)/7
- 2 practical homework assignments (some points), 3 theoretical homework assignments (30 and 40 points respectively).
- Exam can include:
  - Presentation with a discussion of an article
  - Or a conversational oral exam
- Automatic passing grade from 6.5 or higher based on cumulative score (provided all homework assignments are submitted)
- No block grading

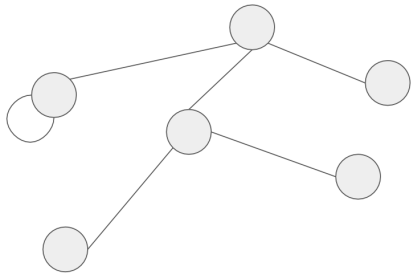
About the Course. Deadlines

- Deadlines are flexible; a penalty of -10% is applied for each day of delay. After three days, the deadline becomes strict
- Twice per course, assignments can be submitted after the strict deadline (an additional 4 days are granted)
- Submissions are not accepted more than 7 days after the deadline
- The original score (without penalties) is considered for grading purposes

About the Course. Homework Policy

- Homework assignments must be completed individually
- No GPT, please

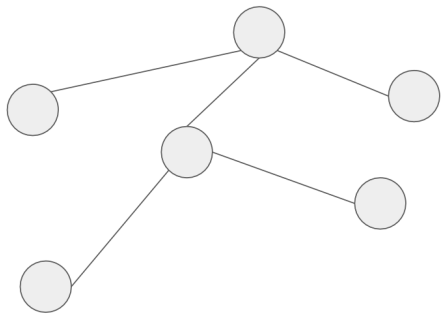
What are Graphs?



What are Graphs?

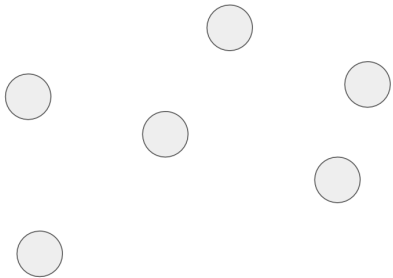
$G = (V, E)$, where:

- V is the set of vertices (or nodes), representing the objects.
- E is the set of edges, representing the relationships or connections between the vertices.



What are Graphs?

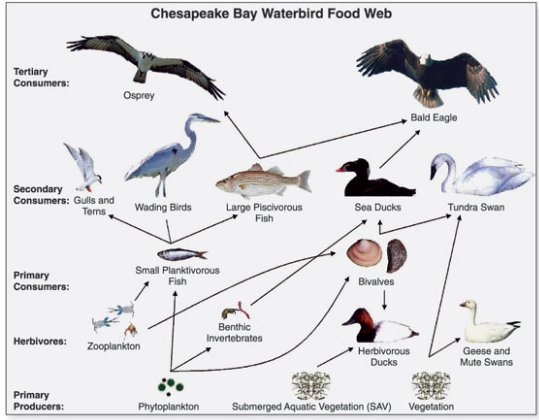
Graph?



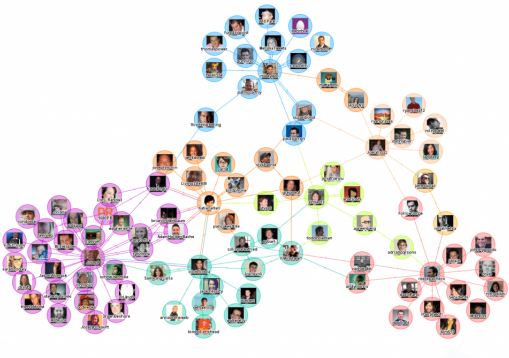
Why Graphs are Needed in the Context of Data?

- Graphs are a generalized way of representing any information related to interactions between objects.
- There are many different methods for graph analysis, making it convenient to transform data from various domains into a unified format.

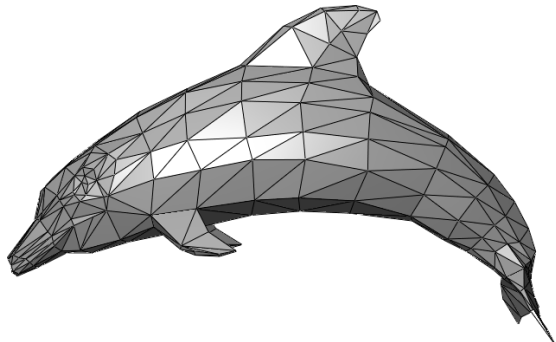
Examples of Graph Data



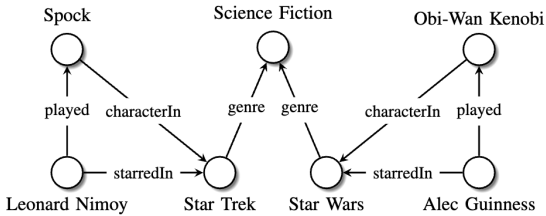
Examples of Graph Data



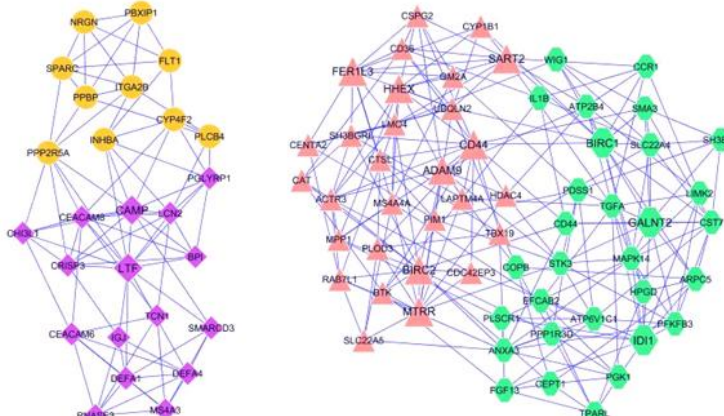
Examples of Graph Data



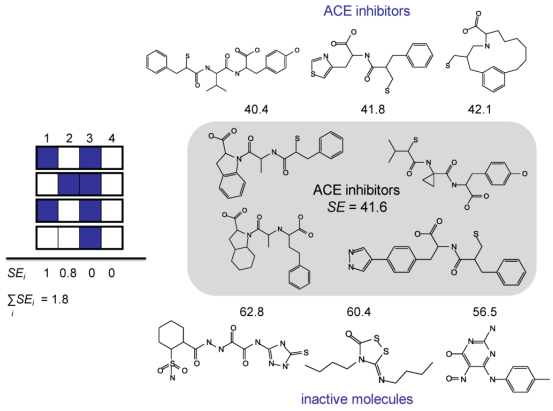
Examples of Graph Data



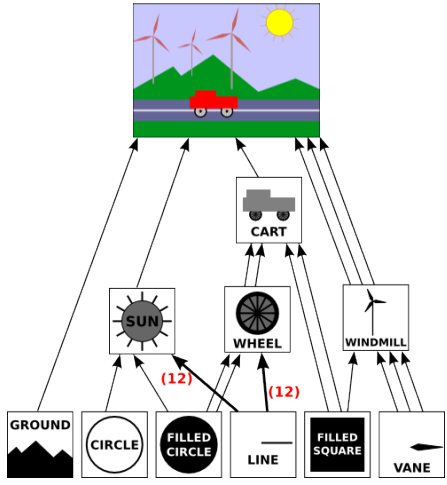
Examples of Graph Data



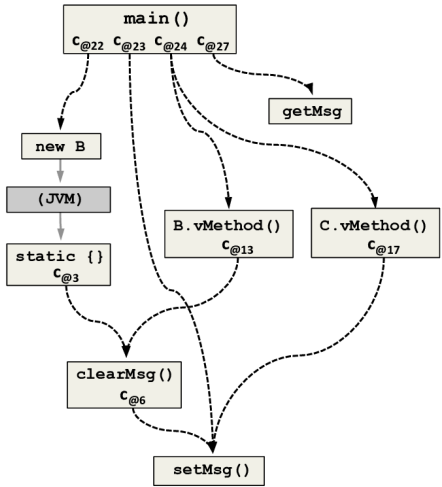
Examples of Graph Data



Examples of Graph Data



Examples of Graph Data



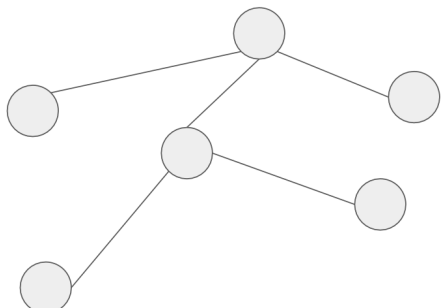
Motivation

The primary applications of deep learning today involve sequential data:

- **NLP:** "askfnkksnf" - sequences of text
- **CV:** Structured data from image pixels
- **Audio:** Sound waves

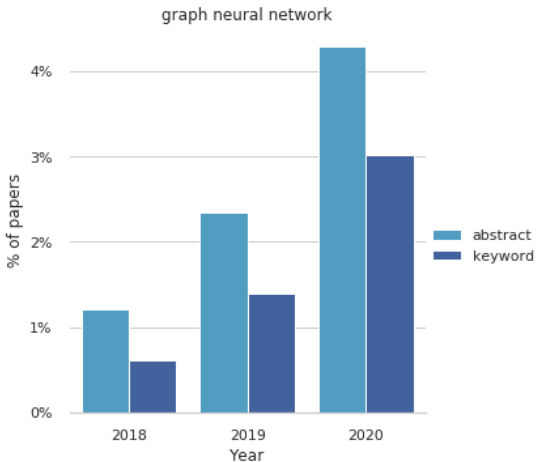
Motivation

The distinctive feature of graphs is the relationships between objects, allowing information to be processed more broadly and differently.



Where is the start and end?

Motivation


graph neural network
4%
3%
2%
1%
0%
% of papers
2018
2019
2020
Year
abstract
keyword

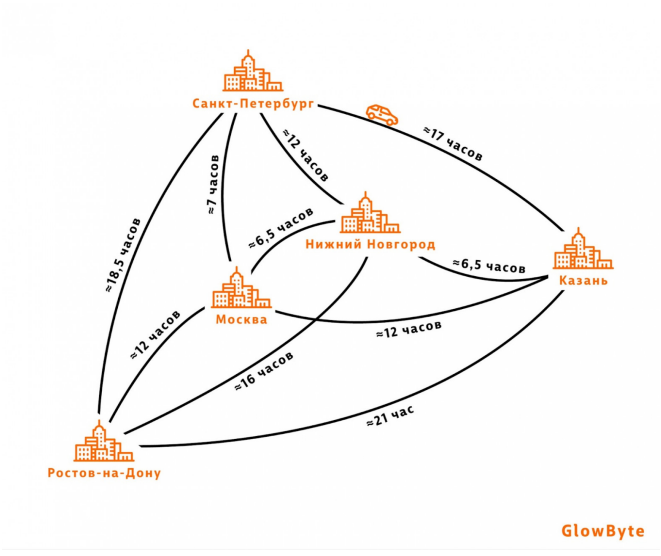
Approaches to Graph Data Analysis Without Neural Networks

Many tasks that are currently solved using deep learning have existed for a long time in the context of graph data. Correspondingly, various methods for solving these tasks using conventional graph algorithms have also existed.

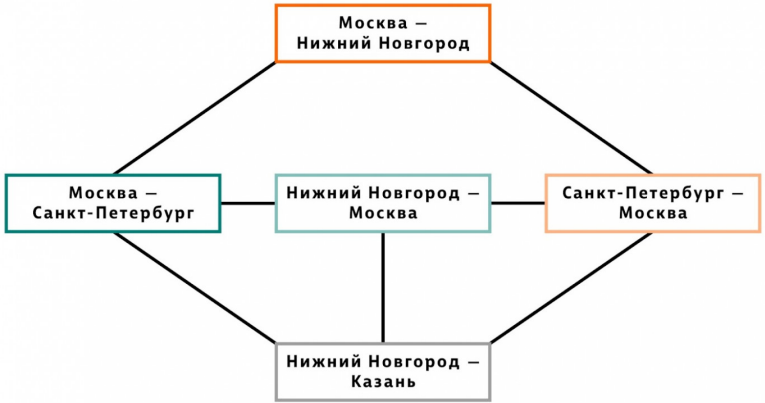
Approaches to Graph Data Analysis Without Neural Networks

- Групповые вкусы пользователя - **начальные вершины обхода графа**
- Делаем 3 случайных шага по графу
- В каждой вершине есть вероятность телепортироваться в начало обхода
- Конечные вершины графа - **кандидаты для рекомендации**

Approaches to Graph Data Analysis Without Neural Networks

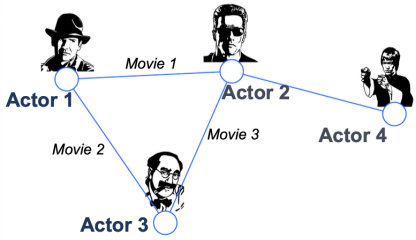


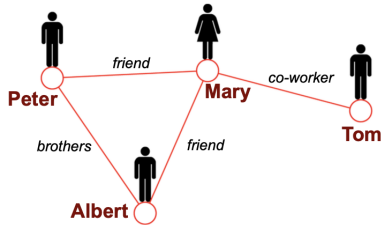
Approaches to Graph Data Analysis Without Neural Networks

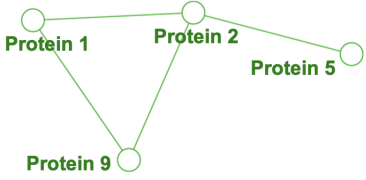


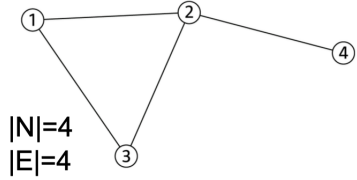
GlowByte

Finally, Graphs!









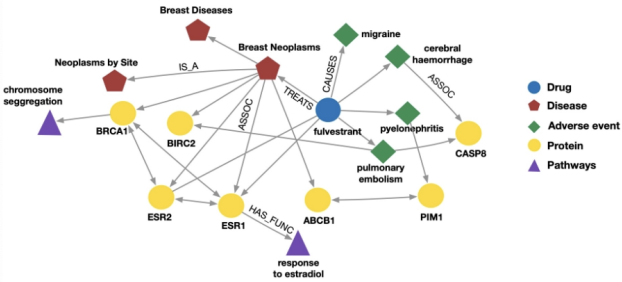
Finally, Heterographs!

A heterogeneous graph is defined as:

$$G = (V, E, R, T)$$

- **Nodes with node types:** $v_i \in V$
- **Edges with relation types:** $(v_i, r, v_j) \in E$
- **Node type:** $T(v_i)$
- **Relation type:** $r \in R$
- **Nodes and edges have attributes/features.**

Finally, Heterographs!



Biomedical Knowledge Graphs

Example node: Migraine
Example edge: (fulvestrant, Treats, Breast Neoplasms)
Example node type: Protein
Example edge type (relation): Causes

Academic Graphs

Example node: ICML
Example edge: (GraphSAGE, NeurIPS)
Example node type: Author
Example edge type (relation): pubYear

Choosing a Proper Representation

How to build a graph?

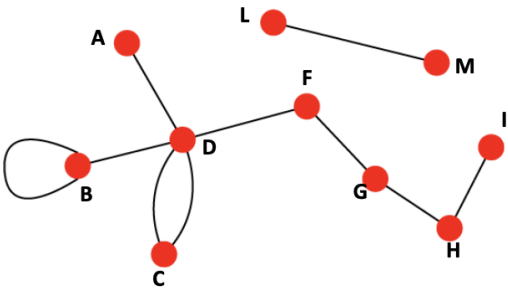
- **What are nodes?**
- **What are edges?**

Choice of the proper network representation of a given domain/problem determines our ability to use networks successfully:

- In some cases, there is a unique, unambiguous representation.
- In other cases, the representation is by no means unique.
- The way you assign links will determine the nature of the question you can study.

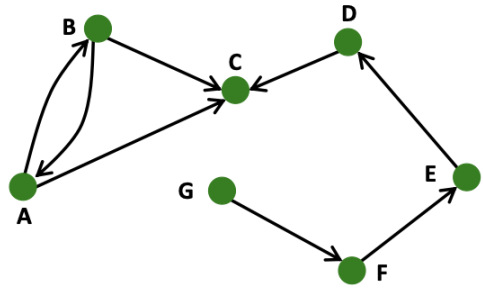
Undirected vs Directed Graphs

- **Links:** undirected (symmetrical, reciprocal)



- Weights
- Properties

- **Links:** directed

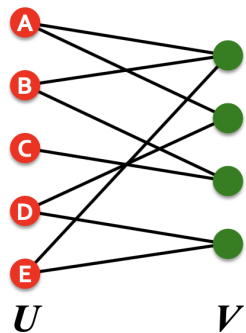


- Types
- Attributes

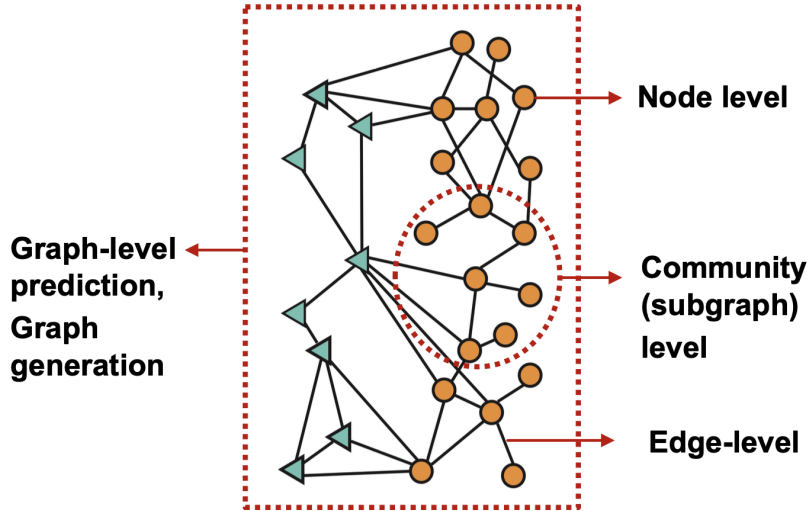
Bipartite graph

Bipartite graph is a graph whose nodes can be divided into two disjoint sets U and V such that every link connects a node in U to one in V; that is, U and V are **independent sets**.

- **Examples:**
  - Authors-to-Papers (they authored)
  - Actors-to-Movies (they appeared in)
  - Users-to-Movies (they rated)
  - Recipes-to-Ingredients (they contain)
- **"Folded" networks:**
  - Author collaboration networks
  - Movie co-rating networks



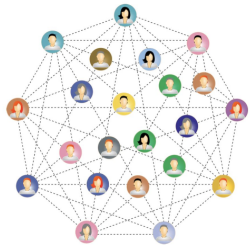
Types of Graph Tasks



Types of Graph Tasks

1. **Edge-level** – Tasks related to edges:
   - Link classification
   - Regression
   - Graph completion
2. **Node-level** – Tasks related to nodes:
   - Node classification
   - Regression
   - Clustering
3. **Graph-level** – Tasks specific to the entire graph:
   - Graph classification
   - Regression
   - Generation
   - Evolution

Node-Level Tasks



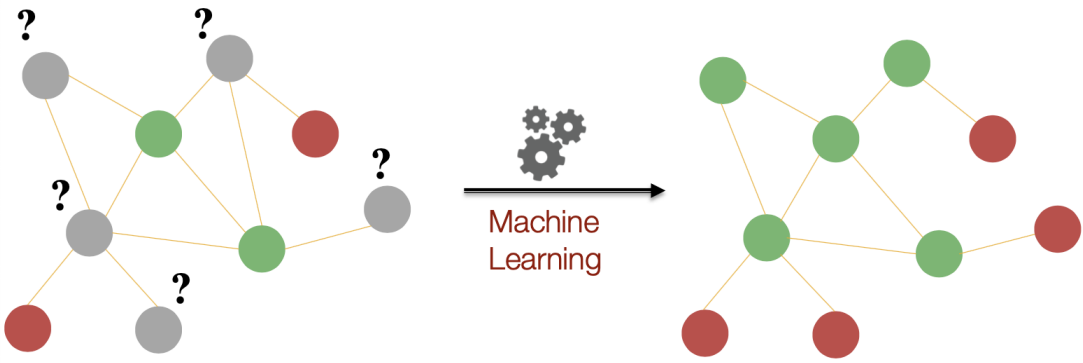
Determining a person's interests based on their surroundings

We aim to learn the function:

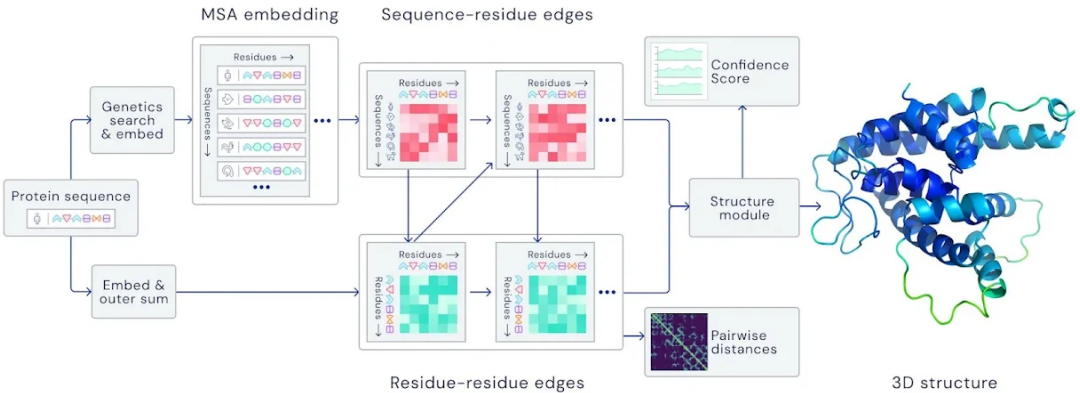
$$f : (V, E) \rightarrow \mathbb{R}^{|V|}$$

where V is the set of nodes, E is the set of edges, and the result is a numerical representation for each node in the graph.

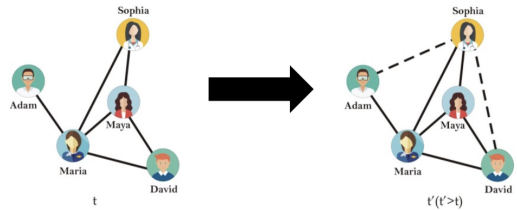
Node-Level Tasks



Node-Level Tasks



Edge-level Tasks



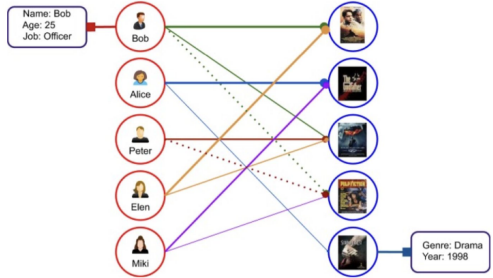
Make friend recommendations in a social network

We aim to make predictions for all pairs (v_i, v_j):

$$f(v_i, v_j)$$

where v_i and v_j are nodes in the graph.

Edge-level Tasks



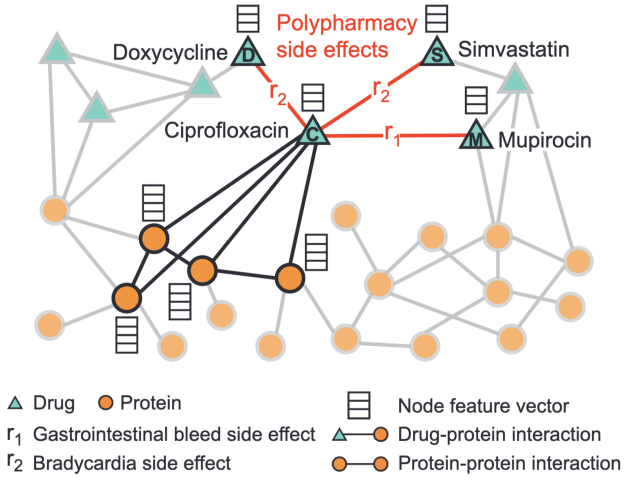
Make recommendations for movies or music

We aim to make predictions for all pairs (v_i, v_j):

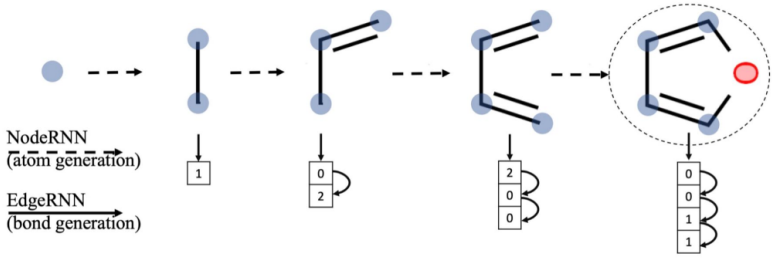
$$f(v_i, v_j)$$

where v_i represents a user and v_j represents an item (e.g., movie or music).

Edge-level Tasks

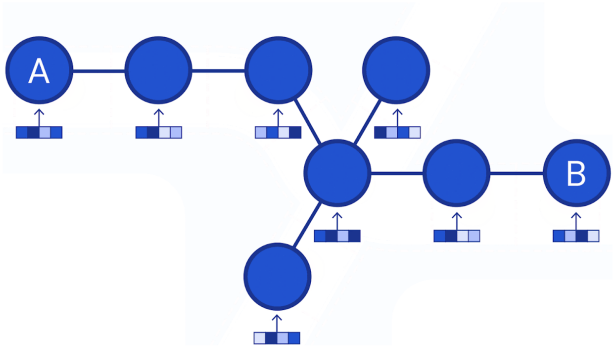


Graph-level Tasks



We aim to learn how to generate graphs with specific characteristics.

Subgraph-level Tasks

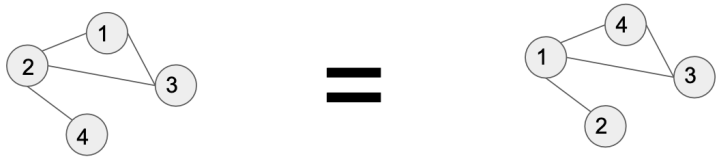


Predicting vehicle arrival time:

- The road is divided into segments (nodes).
- Accessibility between segments (edges) determines connectivity.

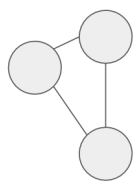
Features of Graph Algorithms

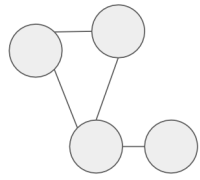
Graph algorithms must operate independently of the permutation of vertex indices.

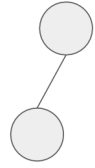


Features of Graph Algorithms

Graph algorithms must be able to handle graphs of different sizes as input.







The End?