Графовые нейросети и их применение

Верещагина Алсу Рашитовна

Сферы:
- Classic ML
- NLP
- GraphNetworks
- TimeSeries
- Uplift modeling

- **DS Sber (Ex Газпромнефть)**
- **Ex преподаватель СПбПУ курса ”Classic ML”**
- **ITMO AI Talent Hub**
- **Победитель и призер 8 хакатонов**

Контакты:
ТГ: @AlsuKrmkv
ТГ: Чат группы

Организационные моменты

	Лекция	Практика
Введение в графы и графовые данные	+	+ HW1
Основы графовых нейросетей	+	+
Ключевые архитектуры GNN	+	+ HW2
Динамические графовые нейросети	+	+
Гетерогенные графы	+	+
Graph Transformers	+	+
Self-supervised Learning	+	+
Применение GNN в индустрии	+	финальный проект

Организационные моменты

Зачетные единицы

1) НМ1 (3 недели)
2) НМ2 (3 недели)
3) Хакатон (3 недели)
4) Экзамен

Система оценивания:

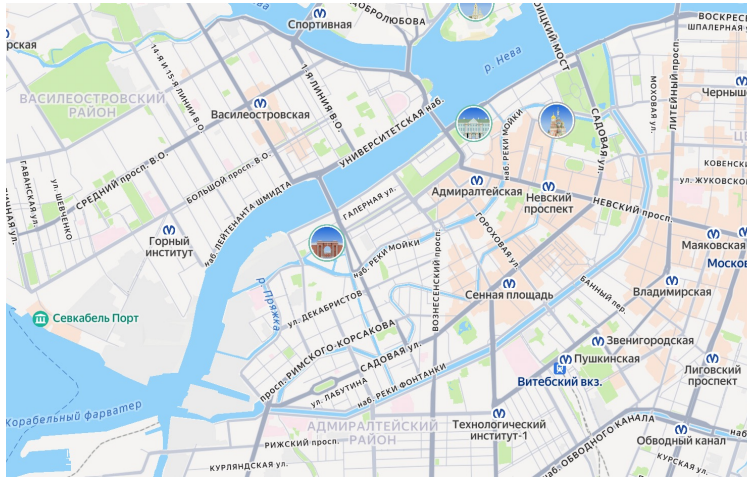
Минимальные требования: 1 домашка + экзамен

«4» - сдать две домашки + повысить оценку можно на экзамене

«9» - сдать две домашки + 1 место в рейтинге с учетом хакатона

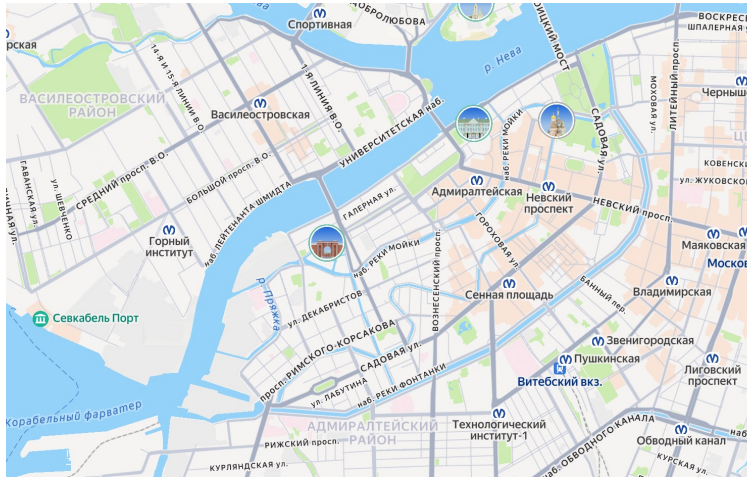
Модуль 1
Введение в графы и графовые данные

Примеры применения графов

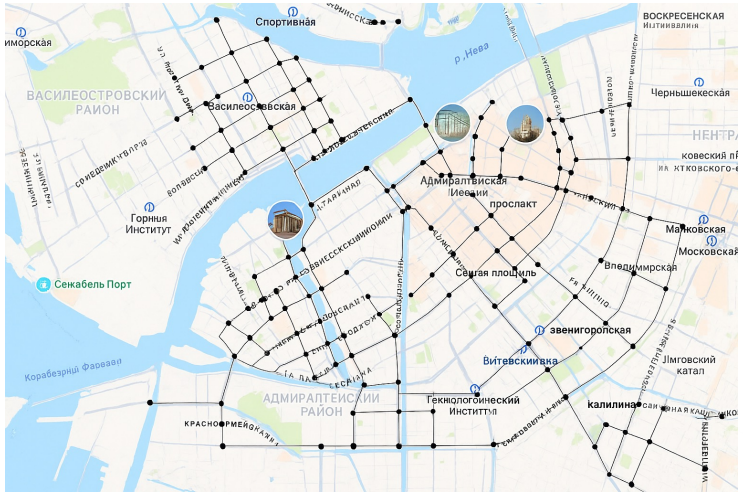


Источник: Яндекс карты

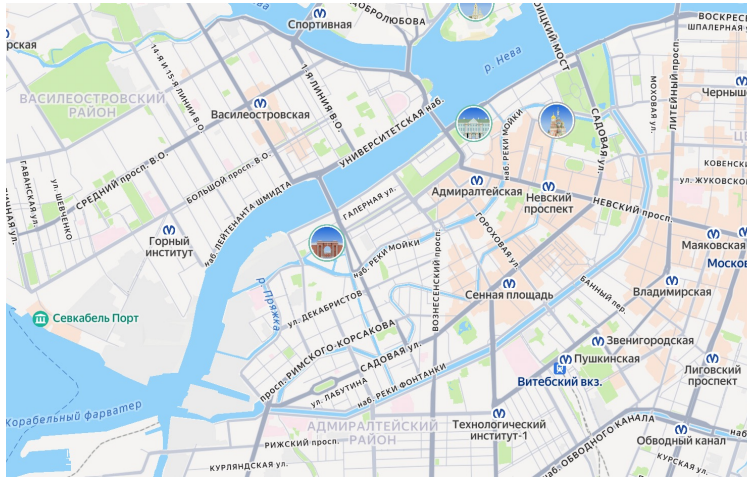
Примеры применения графов

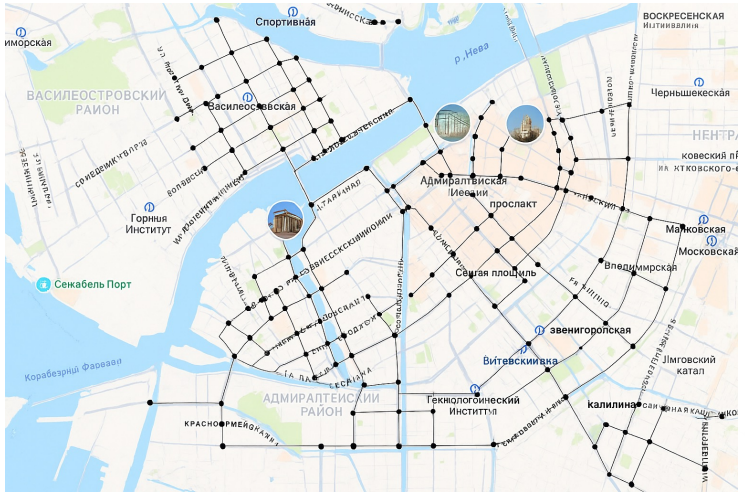


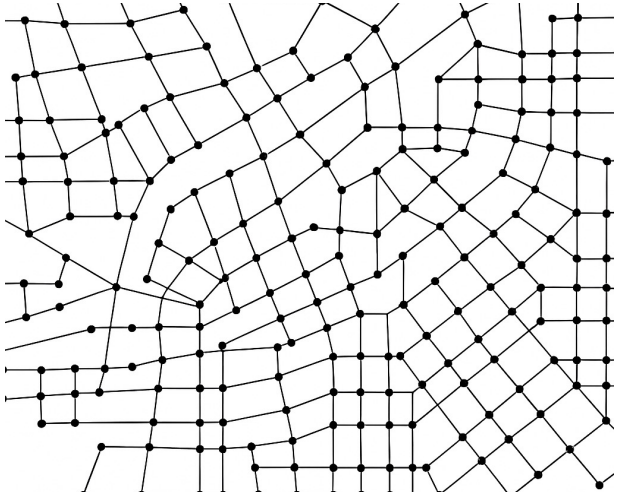
Источник: Яндекс карты



Примеры применения графов



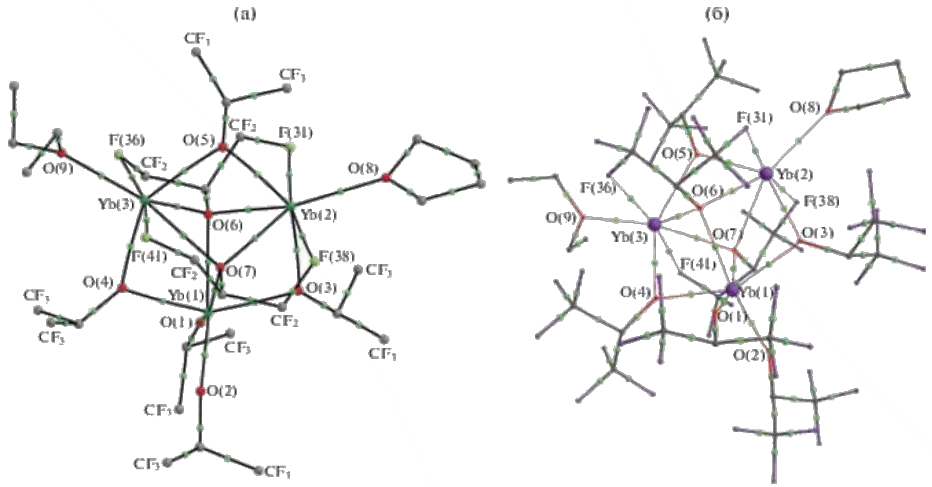




Источник: Яндекс карты

Ссылка на оригинальную статью про использование GNN - https://arxiv.org/pdf/1707.01926

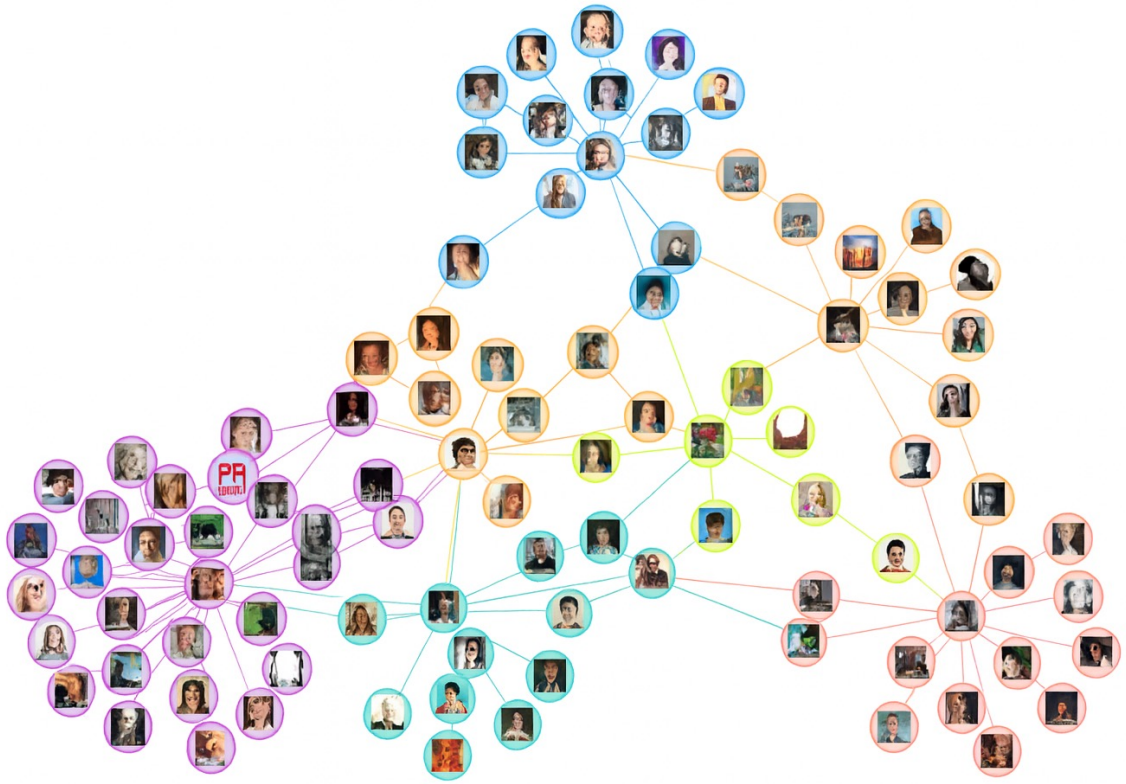
Примеры применения графов



Молекулярный граф

Ссылка на статью - https://www.nature.com/articles/s41586-021-03819-2

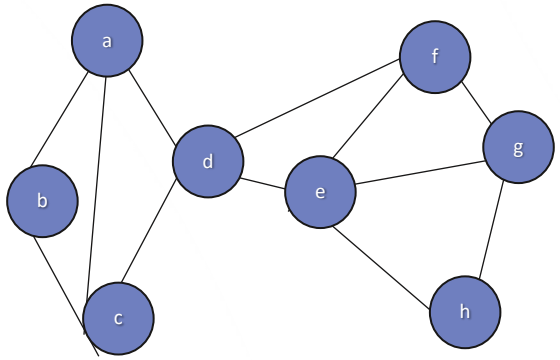
Примеры применения графов



Alibaba GraphRec, TikTok Graph4Rec

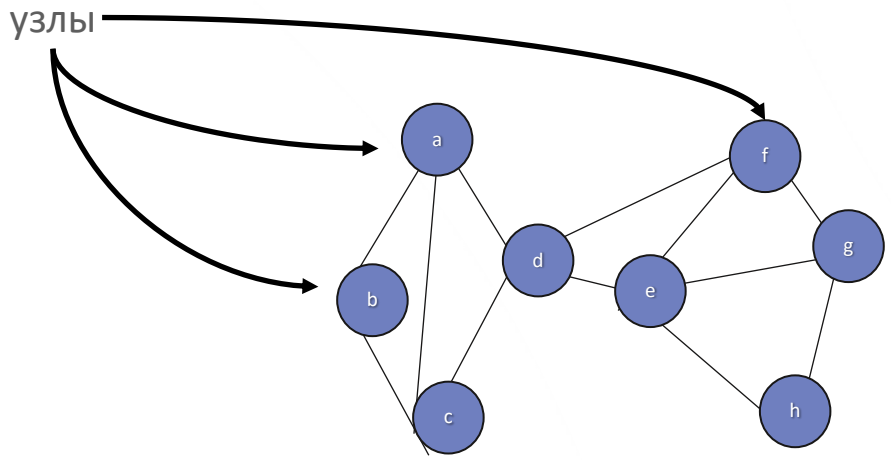
Ссылка на статью PinSage - https://arxiv.org/abs/1806.01973

Основные понятия графов



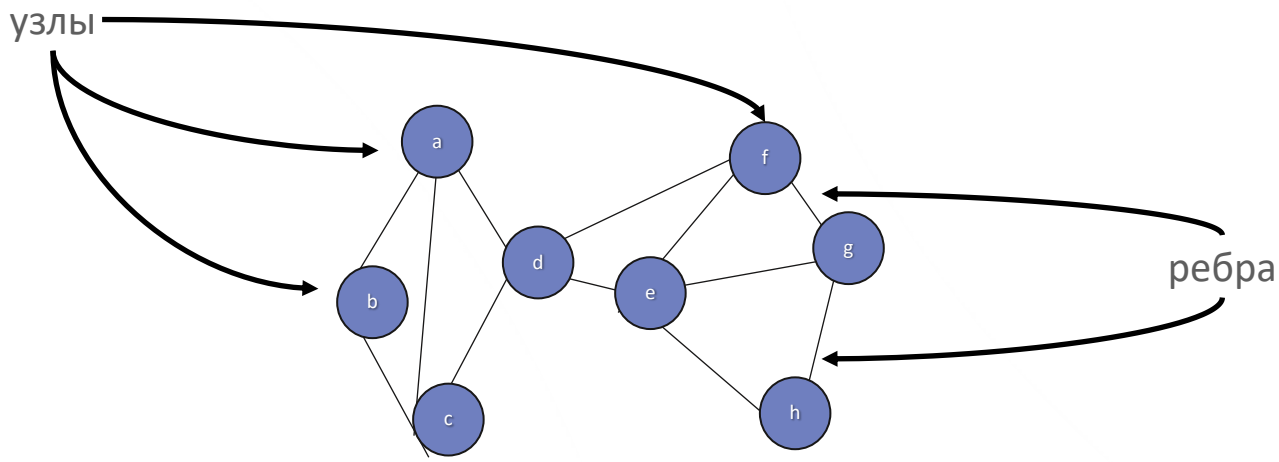
Узлы (вершины) — ?
Ребра (связи) — ?

Основные понятия графов



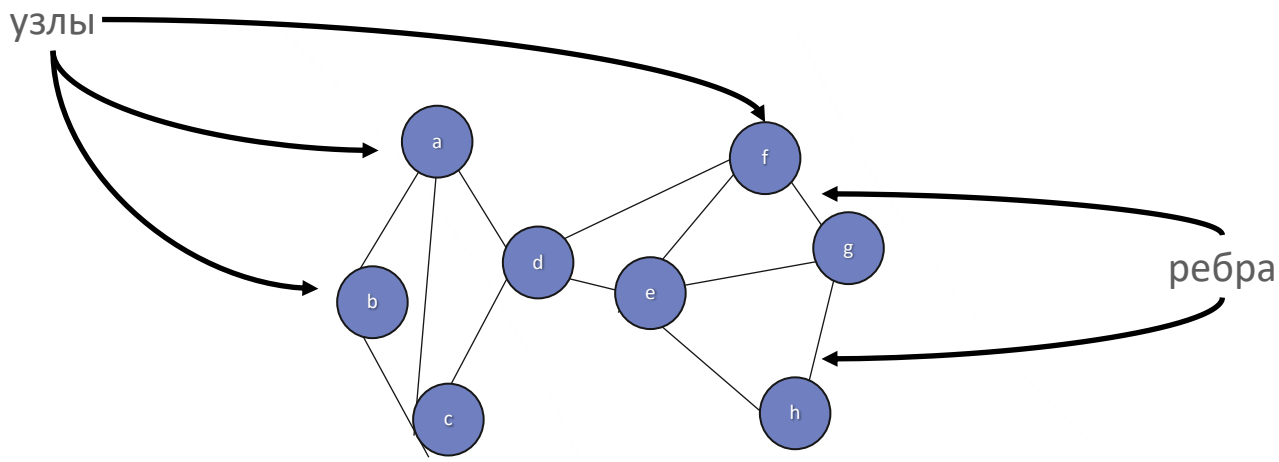
Узлы (вершины) — объекты, которые мы изучаем
Ребра (связи) — ?

Основные понятия графов



Узлы (вершины) — объекты, которые мы изучаем
Ребра (связи) — отношения между объектами

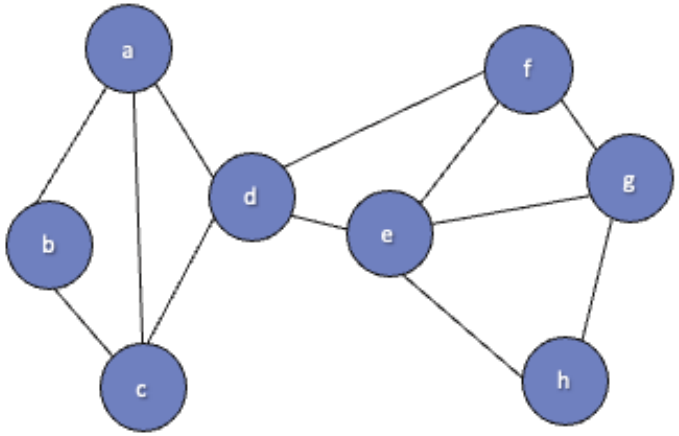
Основные понятия графов

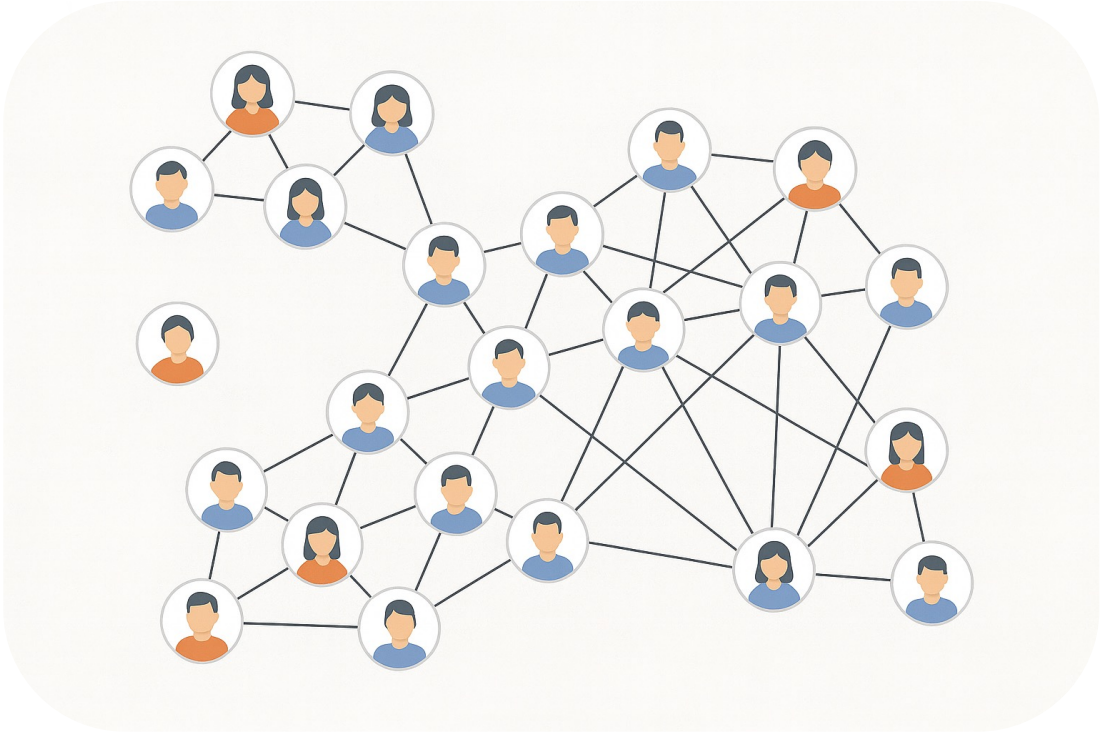


Узлы (вершины) — объекты, которые мы изучаем
Ребра (связи) — отношения между объектами

$G = (V, E)$, где
G — структура, которая состоит их двух множеств
V — множество вершин
E — множество ребер
$|V|$ — количестов вершин
$|E|$ — количество ребер

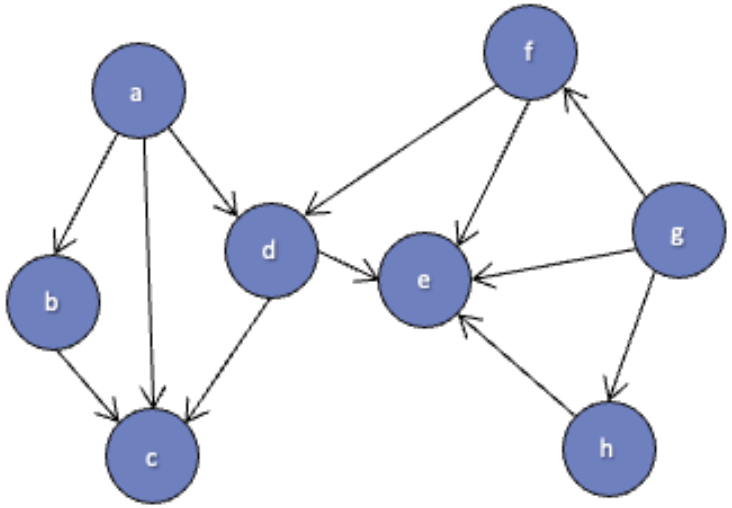
Виды графов. Гомогенные графы



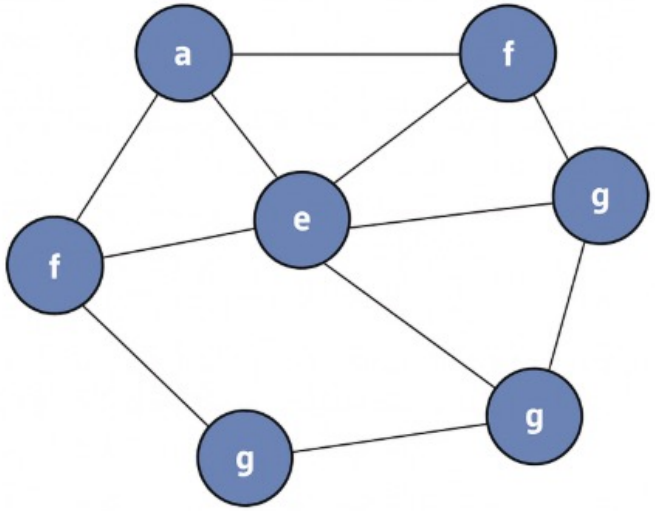


Виды графов по типу направления

Направленный

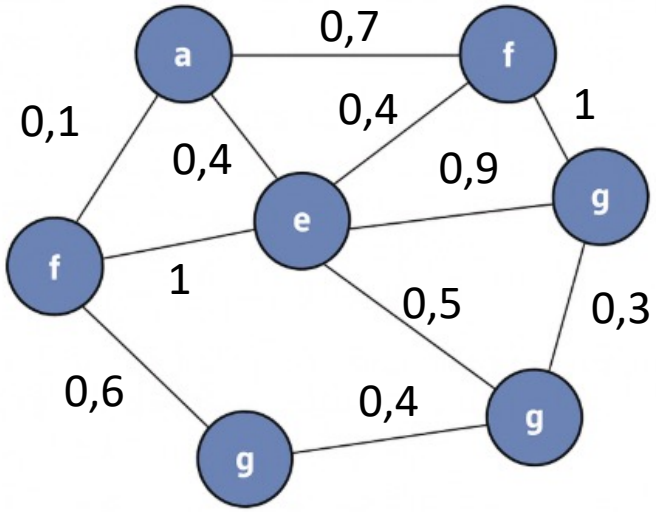


Ненаправленный

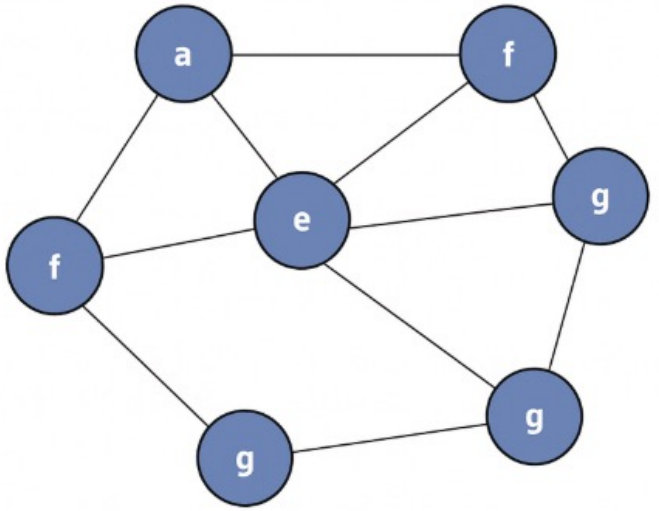


Виды графов по весам

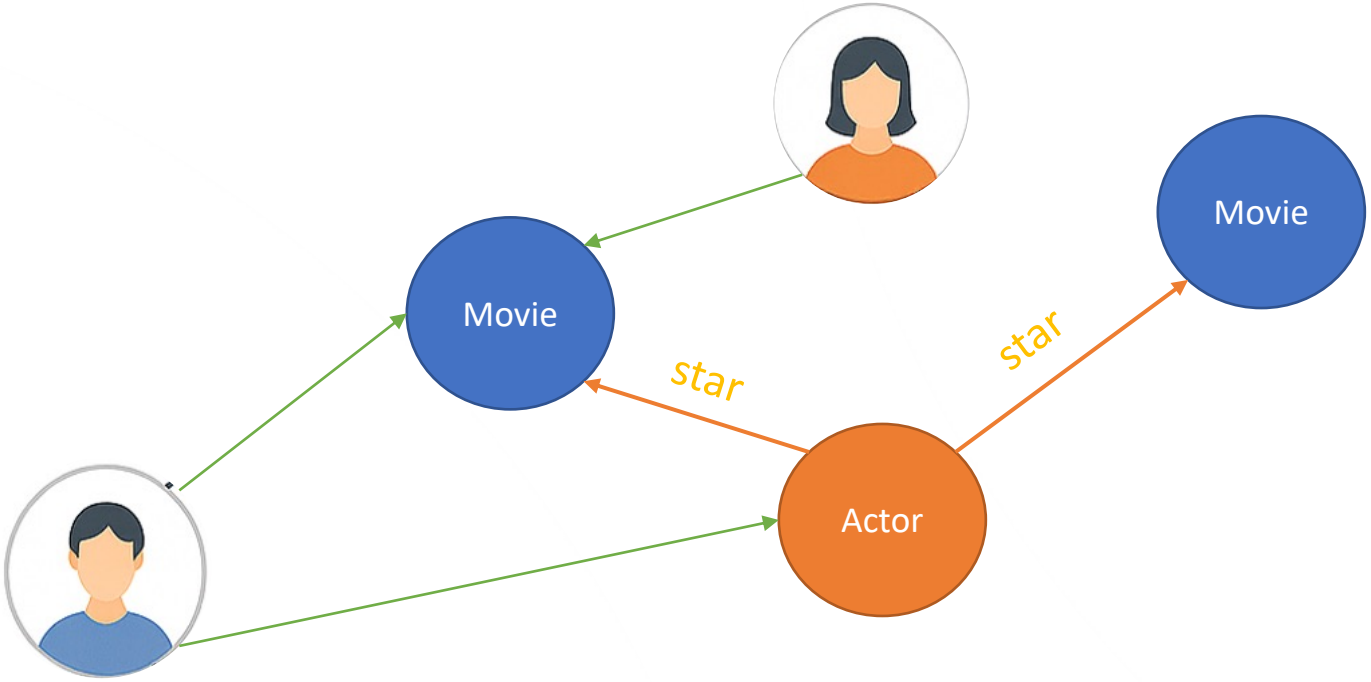
Взвешенный



Невзвешенный



Виды графов. Гетерогенные графы



$G = (V, E, R, T)$, где

G – структура, которая состоит их двух множеств

V – множество вершин и их признаков

E – множество ребер и их признаков

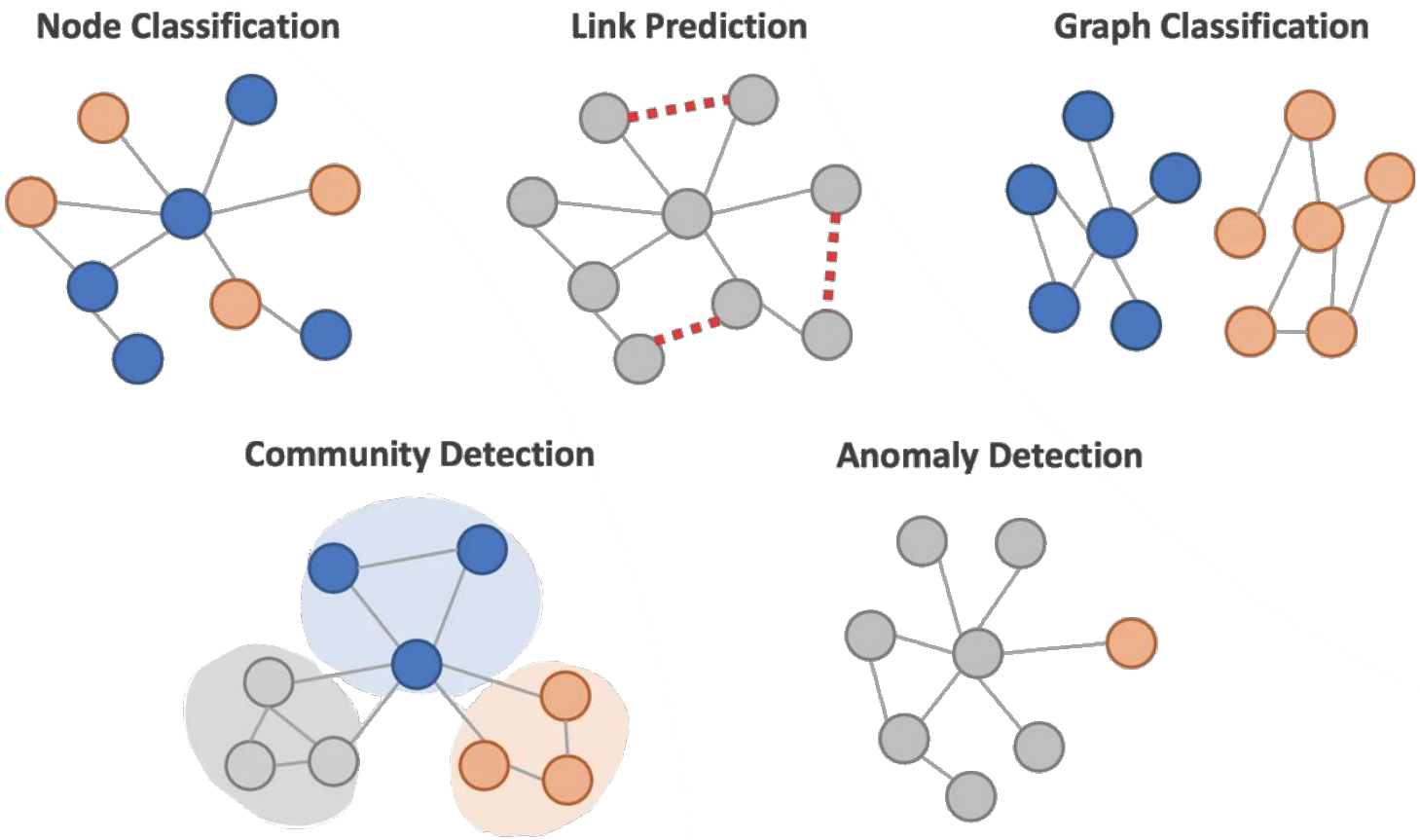
T – можество типов вершин

E – можество типов ребер

Виды разделения графов

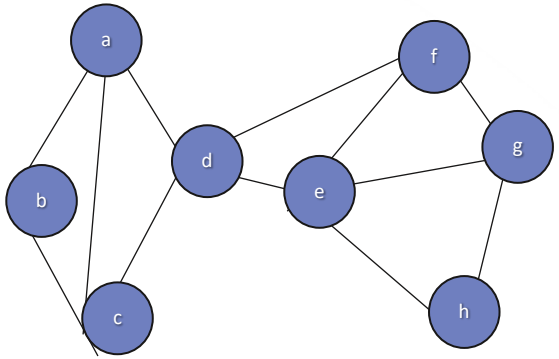
Категория	Виды графов
По направлению	Ориентированные, Неориентированные, Смешанные
По весам	Взвешенные, Невзвешенные
По структуре	Деревья, DAG, Циклические, Планарные, Разреженные, Плотные
По типам узлов/рёбер	Гомогенные, Гетерогенные, Бипартийные, K-partite
По времени	Статические, Динамические, Потоковые, Эволюционирующие
По кратности	Простые, Мультиграфы, Псевдографы
По связности	Связные, Несвязные, k-связные
По размеру	Малые, Средние, Крупные
По природе данных	Социальные, Биологические, Пространственные, Графы знаний, Рекомендательные

Задачи машинного обучения на графах



Источник: https://towardsdatascience.com/graph-neural-networks-with-pyg-on-node-classification-link-prediction-and-anomaly-detection-14aa38fe1275/

Способы описания графов



Матрица смежности

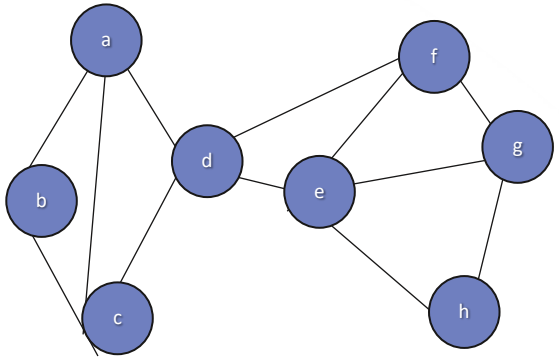
$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Матрица $A \in \mathbb{R}^{|V| \times |V|}$, где:

$$A_{ij} = \begin{cases} 1, & \text{если есть ребро } (i, j) \\ 0, & \text{иначе.} \end{cases}$$

Для **взвешенных графов** в ячейках хранятся веса рёбер w_{ij}.

Способы описания графов



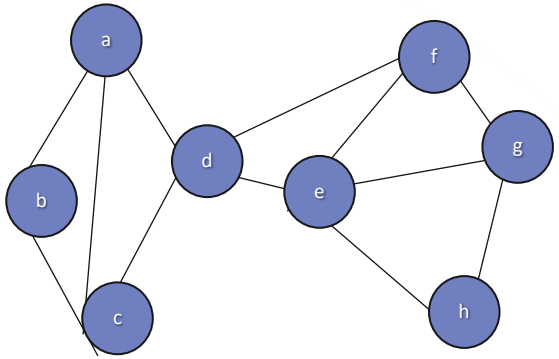
Список смежности (adjacency list)

- a: соседи — $\{b, c, d\}$
- b: соседи — $\{a, c, d\}$
- c: соседи — $\{a, b, d\}$
- d: соседи — $\{a, b, c, e, f\}$
- e: соседи — $\{d, f, g, h\}$
- f: соседи — $\{d, e, g, h\}$
- g: соседи — $\{e, f, h\}$
- h: соседи — $\{e, f, g\}$

Для каждой вершины храним список всех её соседей.

$$Adj[v] = [u_1, u_2, \ldots, u_k]$$

Способы описания графов



Список рёбер (edge list)

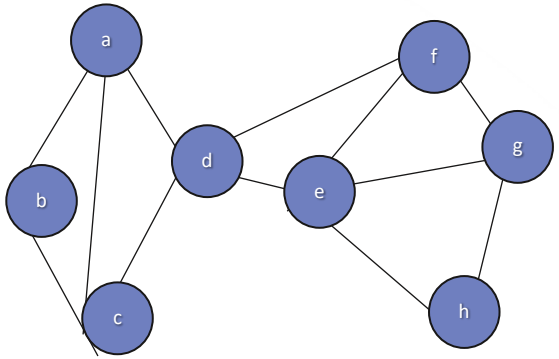
Просто список всех рёбер.

$$E = [(u_1, v_1), (u_2, v_2), ..., (u_m, v_m)]$$

$\{(a,b),(a,c),(a,d),(b,c),(b,d),(c,d),(d,e),(d,f),(e,f),(e,g),(e,h),(f,g),(f,h),(g,h)\}.$

```
edge_index = [
  [a, a, a, b, c, d, d, e, e, e, f, g],
  [b, c, d, c, d, e, f, f, g, h, g, h]
]
```

Способы описания графов

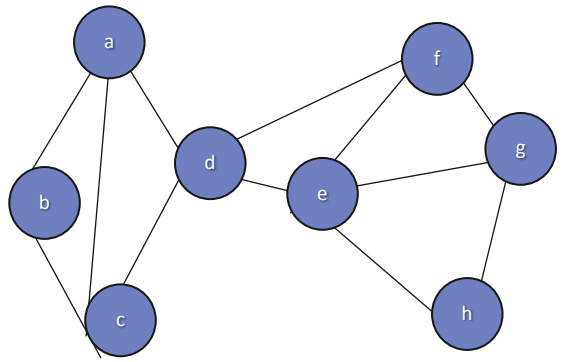


Для каждой вершины храним множество соседей в виде dict

Словарь смежности (Adjacency Dictionary)

```
graph = {
  'a': {'b', 'c', 'd'},
  'b': {'a', 'c'},
  'c': {'a', 'b', 'd'},
  'd': {'a', 'c', 'e', 'f'},
  'e': {'d', 'f', 'g', 'h'},
  'f': {'d', 'e', 'g'},
  'g': {'e', 'f', 'h'},
  'h': {'e', 'g'}
}
```

Способы описания графов



Разреженное хранение: CSR (Compressed Sparse Row)

Храним только ненулевые элементы матрицы смежности.

Для CSR нужно три массива:

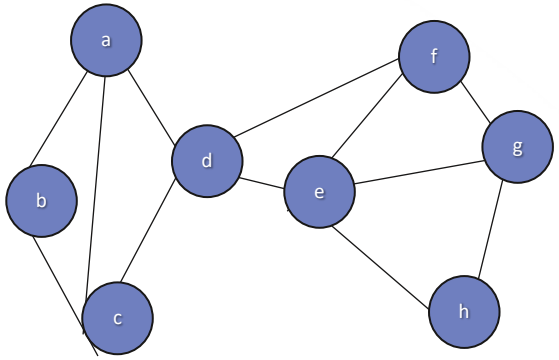
- indices — список всех соседей подряд
- indptr — где начинаются соседи каждой вершины
- data — необязательно (веса)

```
a → [b, c, d]
b → [a, c]
c → [a, b, d]
d → [a, c, e, f]
e → [d, f, g, h]
f → [d, e, g]
g → [e, f, h]
h → [e, g]
```

```
indices = [b, c, d,  a, c,  a, b, d,  a, c, e, f,  d, f, g, h,  d, e, g,  e, f, h,  e, g]
indptr  = [0, 3, 5, 8, 12, 16, 19, 22, 24]
```

indptr[i] = индекс начала соседей i-й вершины

Способы описания графов



COOrdinate Format (COO)

COO (Coordinate) формат — это способ хранить **только ненулевые элементы** матрицы смежности A.

Каждое ненулевое значение (т.е. ребро) представляется **тройкой:**

$$(i, j, A_{ij})$$

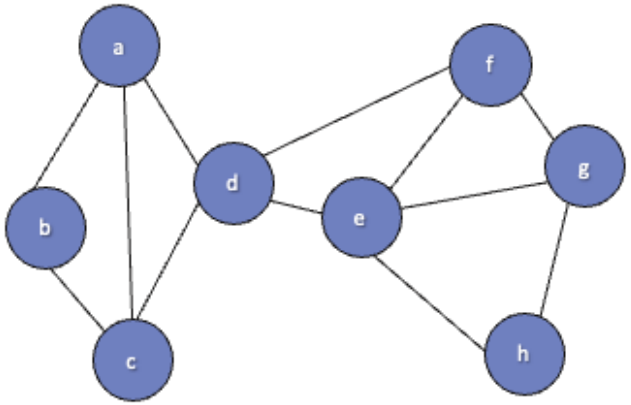
```
row = [a, a, a, b, c, d, d, e, e, e, f, g]
col = [b, c, d, c, d, e, f, f, g, h, g, h]
data = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

Способы описания графов

Метод представления	Память	Проверка ребра	Обход всех рёбер
Матрица смежности (Adjacency Matrix)	$O(n^2)$	$O(1)$	$O(n^2)$
Список смежности (Adjacency List)	$O(n + m)$	$O(k)$	$O(n + m)$
Список рёбер (Edge List)	$O(m)$	$O(m)$	$O(m)$
Словарь смежности (Adjacency Dict)	$O(m)$	$O(1)$	$O(n + m)$
CSR (Compressed Sparse Row)	$O(n + m)$	$O(k)$	$O(n + m)$
COO (Coordinate Format)	$O(m)$	$O(m)$	$O(m)$

n — **количество вершин** (узлов) в графе
m — **количество рёбер** (связей) в графе
k — **степень вершины (**$dev(V)$**)**

Метрики и характеристики графов



Степень вершины

$G = (V, E)$, где

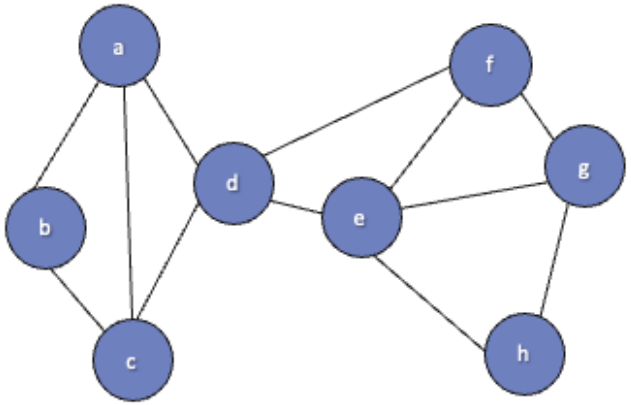
$V -$ множество вершин

$E -$ множество ребер

$|V| -$ количестов вершин

$|E| -$ количество ребер

Метрики и характеристики графов



$G = (V, E)$, где

$V -$ множество вершин

$E -$ множество ребер

$|V| -$ количестов вершин

$|E| -$ количество ребер

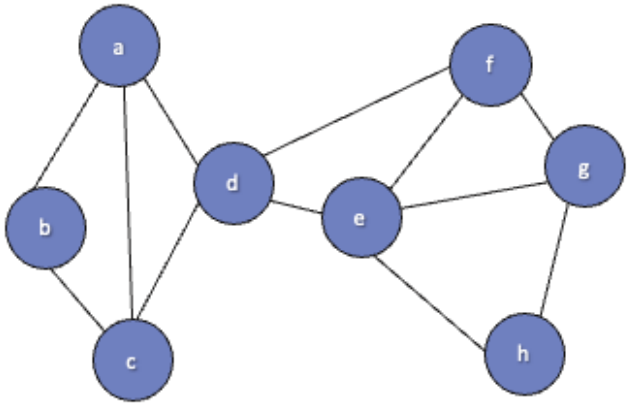
Степень вершины – количество прилежащих ребер

$$deg(v) = |\{u \in V : (v, u) \in E\}|$$

$dev(a) = 3$

$dev(b) = 2$

Метрики и характеристики графов



Плотность графа

$G = (V, E)$, где

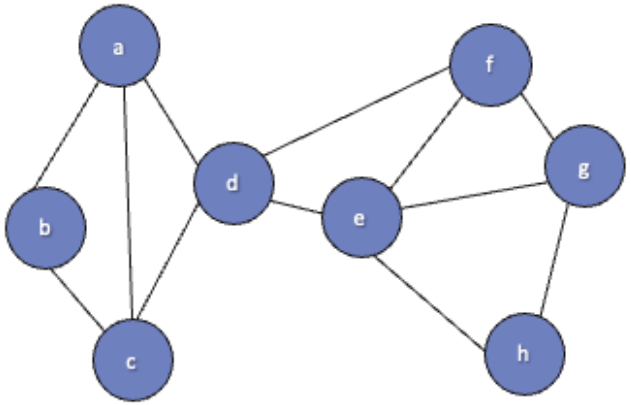
$V -$ множество вершин

$E -$ множество ребер

$|V| -$ количестов вершин

$|E| -$ количество ребер

Метрики и характеристики графов



$G = (V, E)$, где

$V -$ множество вершин

$E -$ множество ребер

$|V| -$ количестов вершин

$|E| -$ количество ребер

Плотность графа – плотностью называется отношение фактического числа рёбер в графе к максимально возможному числу рёбер при данном числе вершин

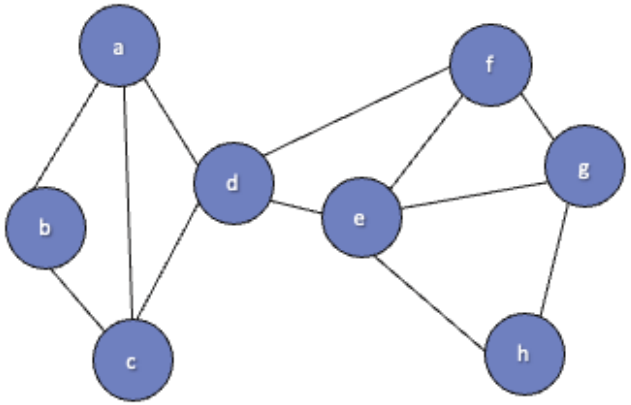
Для неориентированного графа:

$$D = \frac{2m}{n(n-1)}$$

Для ориентированного:

$$D = \frac{m}{n(n-1)}$$

Метрики и характеристики графов



Мера центральности

$G = (V, E)$, где

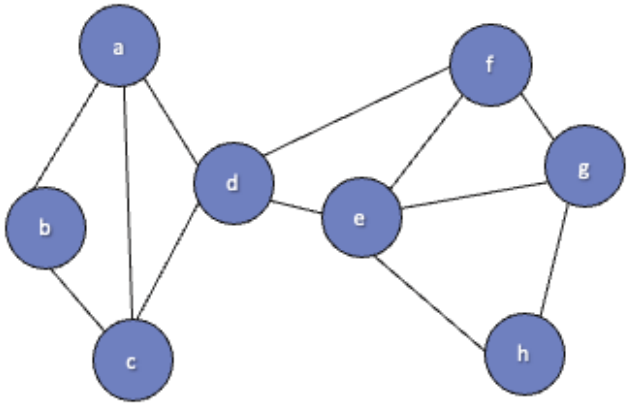
$V -$ множество вершин

$E -$ множество ребер

$|V| -$ количестов вершин

$|E| -$ количество ребер

Метрики и характеристики графов



$G = (V, E)$, где

$V -$ множество вершин

$E -$ множество ребер

$|V| -$ количестов вершин (n)

$|E| -$ количество ребер (m)

Мера центральности - Degree Centrality (Центральность по степени)

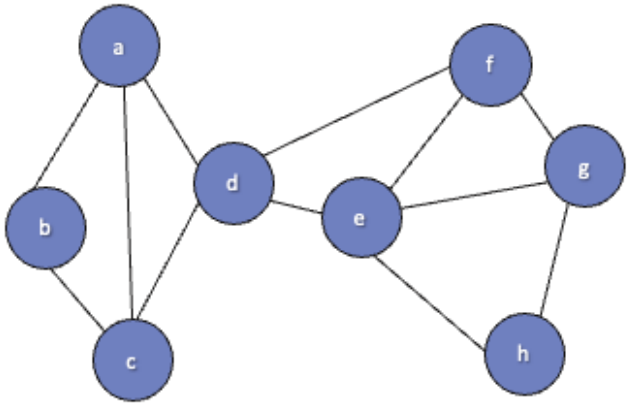
Количество соседей вершины.

$$C_D(v) = deg(v)$$

В нормализованной форме:

$$C_D(v) = \frac{deg(v)}{n-1}$$

Метрики и характеристики графов



$G = (V, E)$, где
$V -$ множество вершин
$E -$ множество ребер
$|V| -$ количестов вершин (n)
$|E| -$ количество ребер (m)

Мера центральности - Closeness Centrality

Показывает, насколько узел **близко к остальным узлам графа** в смысле расстояния (кратчайших путей).

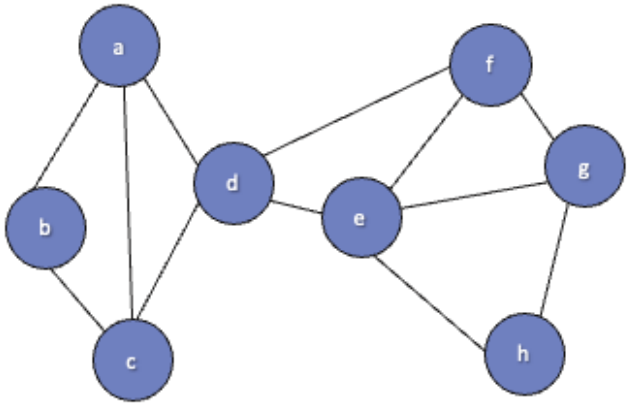
$$C_C(v) = \frac{1}{\sum_{u \neq v} d(v,u)}$$

где $d(v,u)$ — длина кратчайшего пути между v и u.

Иногда нормализуют:

$$C_C(v) = \frac{n-1}{\sum_{u \neq v} d(v,u)}$$

Метрики и характеристики графов



$G = (V, E)$, где
$V -$ множество вершин
$E -$ множество ребер
$|V| -$ количестов вершин (n)
$|E| -$ количество ребер (m)

Мера центральности - Betweenness Centrality

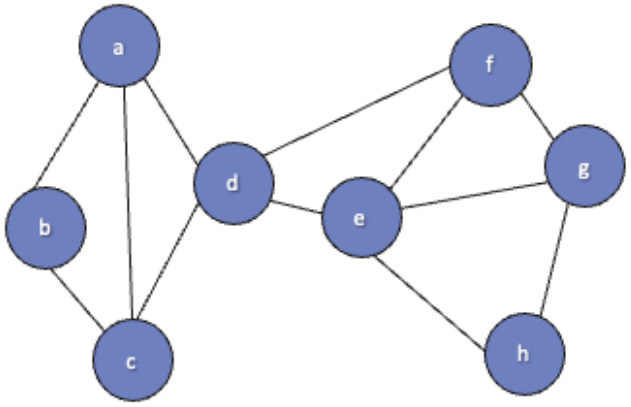
Показывает, **насколько часто вершина лежит на кратчайших путях между другими парами**.

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

где

- σ_{st} — количество кратчайших путей между s и t,
- $\sigma_{st}(v)$ — из них проходят через v.

Метрики и характеристики графов



$G = (V, E)$, где

$V -$ множество вершин

$E -$ множество ребер

$|V| -$ количестов вершин (n)

$|E| -$ количество ребер (m)

Мера центральности - Eigenvector Centrality

“Узел важен, если он связан с другими важными узлами.”

Формально — компонент собственного вектора матрицы смежности:

$$Ax = \lambda x$$

где

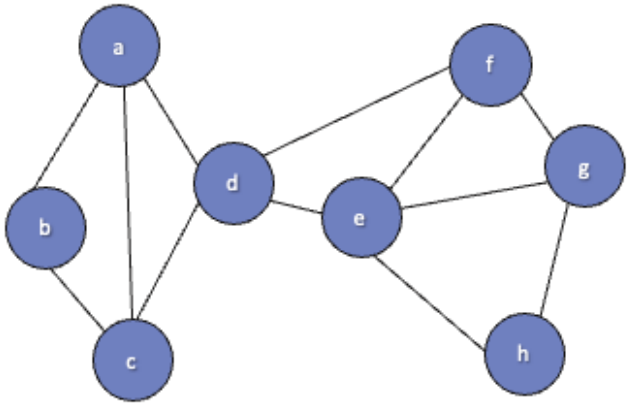
A — матрица смежности,

x_i — центральность узла i,

λ — собственное значение.

$$C_E(v_i) = \frac{1}{\lambda} \sum_j A_{ij} C_E(v_j)$$

Метрики и характеристики графов



$G = (V, E)$, где

$V -$ множество вершин

$E -$ множество ребер

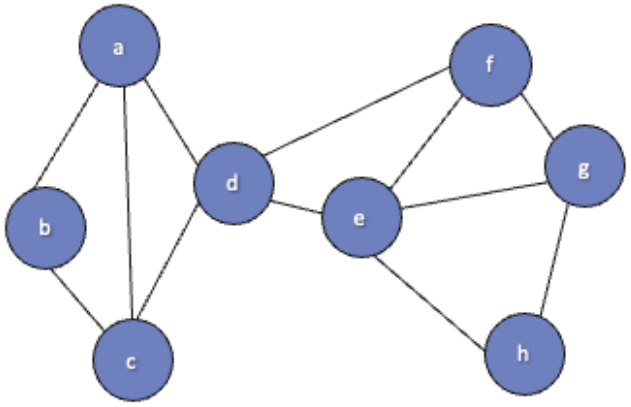
$|V| -$ количестов вершин (n)

$|E| -$ количество ребер (m)

Мера центральности - PageRank

PageRank — это мера влияния или авторитетности узла в ориентированном графе.
Она показывает, насколько “важен” узел, исходя из того, сколько и какие другие важные узлы на него ссылаются.

Метрики и характеристики графов



$G = (V, E)$, где

V – множество вершин

E – множество ребер

$|V|$ – количестов вершин (n)

$|E|$ – количество ребер (m)

Мера центральности - PageRank

Для графа с n узлами:

$$PR(v_i) = \alpha \sum_{v_j \in N_{in}(v_i)} \frac{PR(v_j)}{deg_{out}(v_j)} + (1-\alpha)\frac{1}{n}$$

где:

- $PR(v_i)$ — рейтинг вершины i,
- $N_{in}(v_i)$ — множество вершин, у которых есть ссылка на v_i,
- $deg_{out}(v_j)$ — число исходящих ссылок у узла v_j,
- $\alpha \in (0, 1)$ — коэффициент затухания (обычно 0.85),
- $\frac{1-\alpha}{n}$ — вероятность “прыжка” к случайной вершине.

Example (3): Physics Simulation

Ссылки и источники

1. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting - https://arxiv.org/pdf/1707.01926
2. Highly accurate protein structure prediction with AlphaFold - https://www.nature.com/articles/s41586-021-03819-2
3. Graph Convolutional Neural Networks for Web-Scale Recommender Systems - https://arxiv.org/abs/1806.01973
4. DeepMind & Google’s ML-Based GraphCast Outperforms the World’s Best Medium-Range Weather Forecasting System - https://www.science.org/stoken/author-tokens/ST-1550/full
5. Learning to Simulate Complex Physics with Graph Networks - https://arxiv.org/pdf/2002.09405
6. Page Rank - http://ilpubs.stanford.edu:8090/422/?utm_campaign=Technical%20SEO%20Weekly&utm_medium=ema