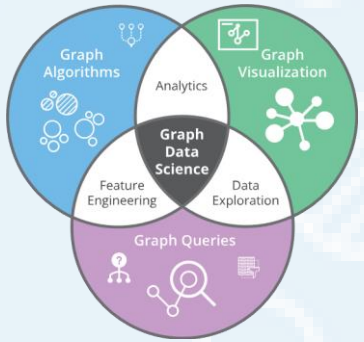


Habib University
shaping futures

CS343 Graph Data Science

Node Similarity

Muhammad Qasim Pasta
qasim.pasta@sse.habib.edu.pk

Node Similarity

- Measuring how "similar" two nodes are in a network based on their structure.
- Node similarity measures are numerical values used to quantify the similarity or dissimilarity between two nodes in a network.

Node Similarity Measures

- Node similarity measures are numerical values used to quantify the similarity or dissimilarity between two nodes in a network.

Applications:

- Social Media Friend Recommendations (Facebook, LinkedIn, Twitter)
  - Platforms suggest new friends/connections based on mutual friends (common neighbors) and interaction patterns.
  - If Alice and Bob have many mutual friends, Facebook might suggest that Alice sends Bob a friend request.

Applications (cont…)

- Movie & Music Recommendation (Netflix, Spotify, YouTube)
  - Recommender systems suggest movies or songs by comparing users with similar preferences.
  - If two users watch the same set of movies, Netflix will recommend a movie watched by one but not the other.

- E-commerce Product Recommendations (Amazon, eBay, Shopee)
  - When a user views or buys a product, similar products are recommended based on browsing/purchase history.
  - If many users who bought a laptop also bought a wireless mouse, Amazon will suggest a mouse when someone buys a laptop.

Node Similarity Measures

- Node similarity measures are numerical values used to quantify the similarity or dissimilarity between two nodes in a network
- Jaccard Index:
  - common neighbors

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

- Overlap Coefficient
  - measures the ratio of the intersection to the smaller set

$$O(A,B) = \frac{|A \cap B|}{min(|A|,|B|)}$$

- Cosine Similarity
  - If nodes are represented by their **neighbor sets** (adjacency lists), cosine similarity measures the structural overlap.

$$cos_w(A,B) = \sum_i \frac{\alpha_i \cdot \beta_i}{\sqrt{\sum_i \alpha_i^2} \cdot \sqrt{\sum_i \beta_i^2}}.$$

Node Similarity:

- Syntax:

CALL gds.nodeSimilarity.stream(
graphName: String,
configuration: Map)
YIELD node1: Integer, node2: Integer, similarity: Float

similarityMetric	String	`JACCARD`	yes	The metric used to compute similarity. Can be either `JACCARD`, `OVERLAP` or `COSINE`.

call gds.nodeSimilarity.stream("purchases",{**similarityMetric:"JACCARD"})** yield node1, node2, similarity
Return *

Comparing Nodes

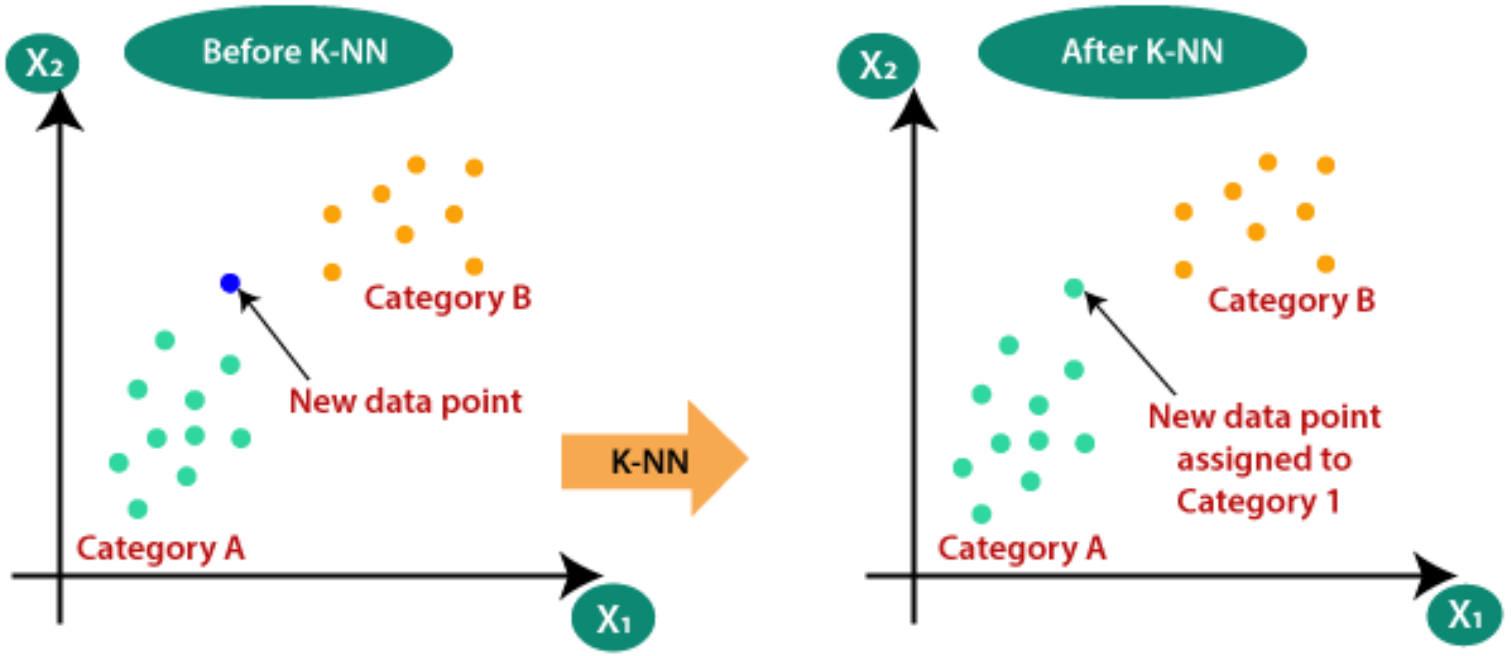
```
call gds.nodeSimilarity.stream("purchases",{similarityMetric:"COSINE"}) yield node1, node2, similarity
with gds.util.asNode(node1).name as from, gds.util.asNode(node2).name as to, similarity
WHERE similarity > 0.5
return from,to,similarity
ORDER BY similarity DESC
```

K-Nearest Neighbour

- KNN is a lazy learning algorithm (no explicit training phase).
- It is a non-parametric method (does not assume any underlying data distribution).
- Used for classification and regression tasks.

Working

- Given a data point, find its K nearest neighbors using a distance metric.
- The label (classification) or average (regression) of the neighbors determines the output.



K Nearest Neighbour (KNN)

- Supervised machine learning algorithm used for classification and regression tasks.

Choosing the right K

- Small K → Sensitive to noise, high variance (overfitting).
- Large K → Smoother decision boundary, but risk of underfitting.
- Rule of Thumb: $K = \sqrt{N}$ (where N is the number of samples).