Course code : **CSE3009**
Course title : **No SQL Data Bases**
Module : **6**
Topic : **6**

Link Analysis Algorithms

Objectives

This session will give the knowledge about

- Link analysis algorithms
- Web as a Graph
- Topic specific page rank

Introduction to Link analysis

Link Analysis deals with mining useful information from linked structures like graphs.

Graphs have vertices representing objects and links among those vertices representing relationships among those objects.

The most common interpretation of the word link today is hyperlink - a means of connecting two web documents wherein activating a special element embedded in one document takes you to the other.

Introduction to Link analysis

A link represents a relationship and connects two objects that are related to each other in that specific way.

A collection of links representing the same kind of relationship form a network, or graph, where the objects being related correspond to the graph vertices and the links themselves are the edges.

When two objects being related by a link are of the same kind, then the network formed by such links is termed a homogeneous network.

Introduction to Link analysis

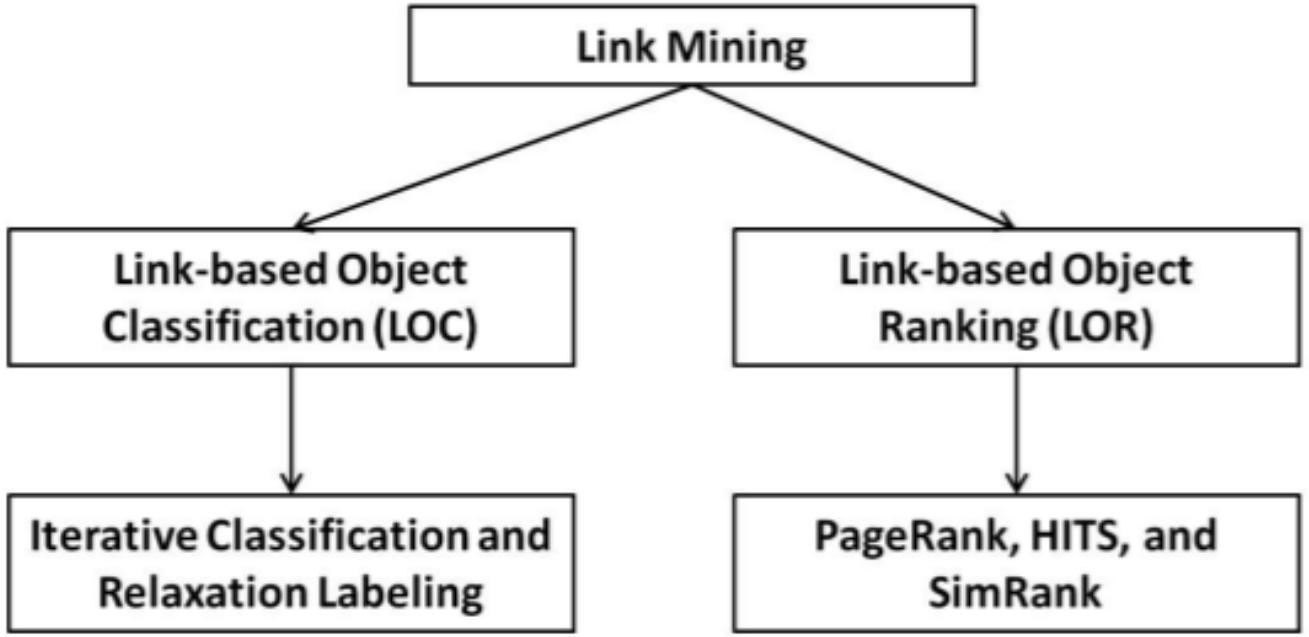
The friendship relation, for instance, forms a homogeneous network of friends, whereas the car-owner relation defines a heterogeneous network.

When a network consists of several kinds of links, it is said to be multi-relational, or sometimes multi-mode.

An example could be a family tree that connects people using relationships, such as parent-child, sibling, spouse, etc.

Link analysis can also give us some interesting insight into the world around us. In the World Wide Web, if web pages are considered as nodes and hyperlinks as edges, then the link analysis will help us to understand which page is appropriate.

Kinds of Link Analysis Tasks



Link-based Object Classification (LOC)

LOC (Link-based Object Classification) is a technique used to assign class labels to nodes according to their link characteristics. One very simplified example is to classify nodes as strongly connected and weakly connected depending solely on their degree.

LOC can also incorporate information about a node's properties for classification.

For instance, if your task is to create compatible teams from a pool of personnel, and you have generic preference data from everyone, then you can build up a graph, where each node represents a person and each edge represents a common preference between two persons

A slightly more complex process would be to find the average distance of each node to all the other nodes, and classify them according to that quantity.

Link-based Object Ranking (LOR)

LOR (Link-based Object Ranking) ranks objects in a graph based on several factors affecting their importance in the graph structure, whereas LOC assigns labels specifically belonging to a closed set of finite values to an object.

The purpose of LOR is not to assign distinctive labels to the nodes—usually, all nodes in such networks are understood to be of the same type—the goal is to associate a relative quantitative assessment with each node.

LOR can sometimes be a more fine-grained version of LOC, such as in our strongly/weakly connected example, if we desire to mark each node with the precise number representing its degree of connectivity, then it can be one form of ranking the nodes.

Link-based Object Ranking (LOR)

Ranking tasks are usually much more complex than that, and take into account a large part of the graph when coming up with a figure for each node.

One of the most well-known ranking tasks is ranking web pages according to their relevance to a search query.

Link prediction

While LOC and LOR use analysis of links to talk about the nodes in a network, Link Prediction actually deals with links themselves.

A common example of prediction is trying to guess which authors will co-author a paper in the future, given a current collaboration graph.

Here, we try to infer new collaboration links, but do not say anything new about the authors themselves.

Link prediction

Some Link prediction Algorithms:

- Adamic Adar (algo.linkprediction.adamicAdar)
- Common Neighbors (algo.linkprediction.commonNeighbors)
- Preferential Attachment (algo.linkprediction.preferentialAttachment)
- Resource Allocation (algo.linkprediction.resourceAllocation)
- Same Community (algo.linkprediction.sameCommunity)
- Total Neighbors (algo.linkprediction.totalNeighbors)

Common Neighbours

Common neighbors captures the idea that two strangers who have a friend in common are more likely to be introduced than those who don't have any friends in common.

It is computed using the following formula:

common neighbors $$CN(x, y) = |N(x) \cap N(y)|$$

where N(x) is the set of nodes adjacent to node x, and N(y) is the set of nodes adjacent to node y. A value of 0 indicates that two nodes are not close, while higher values indicate nodes are closer.

The library contains a function to calculate closeness between two nodes.

Common Neighbours algorithm sample

```
MERGE (zhen:Person {name: "Zhen"})

MERGE (praveena:Person {name: "Praveena"})

MERGE (michael:Person {name: "Michael"})

MERGE (arya:Person {name: "Arya"})

MERGE (karin:Person {name: "Karin"})
```

Common Neighbors algorithm sample

MERGE (zhen)-[:FRIENDS]-(arya)

MERGE (zhen)-[:FRIENDS]-(praveena)

MERGE (praveena)-[:WORKS_WITH]-(karin)

MERGE (praveena)-[:FRIENDS]-(michael)

MERGE (michael)-[:WORKS_WITH]-(karin)

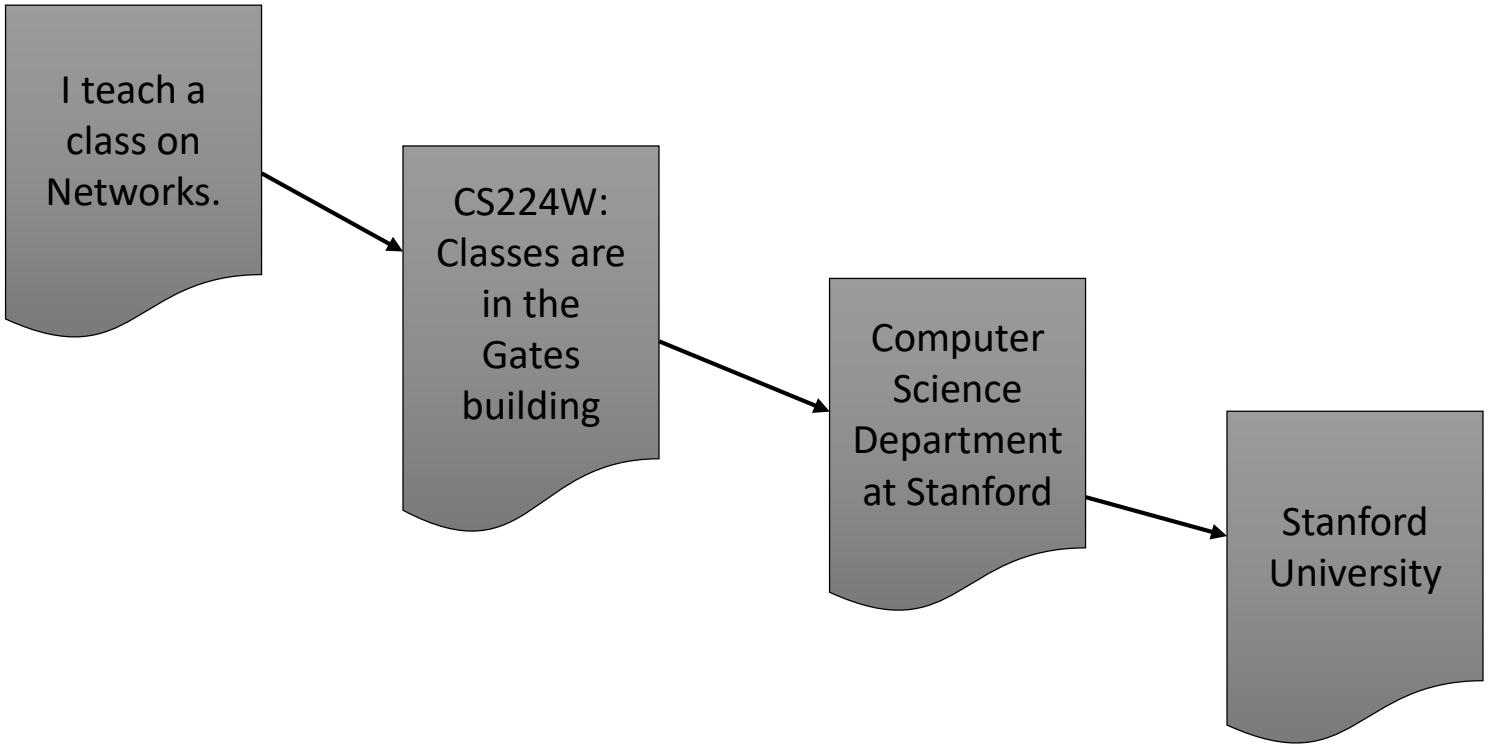
MERGE (arya)-[:FRIENDS]-(karin)

Common Neighbors algorithm sample

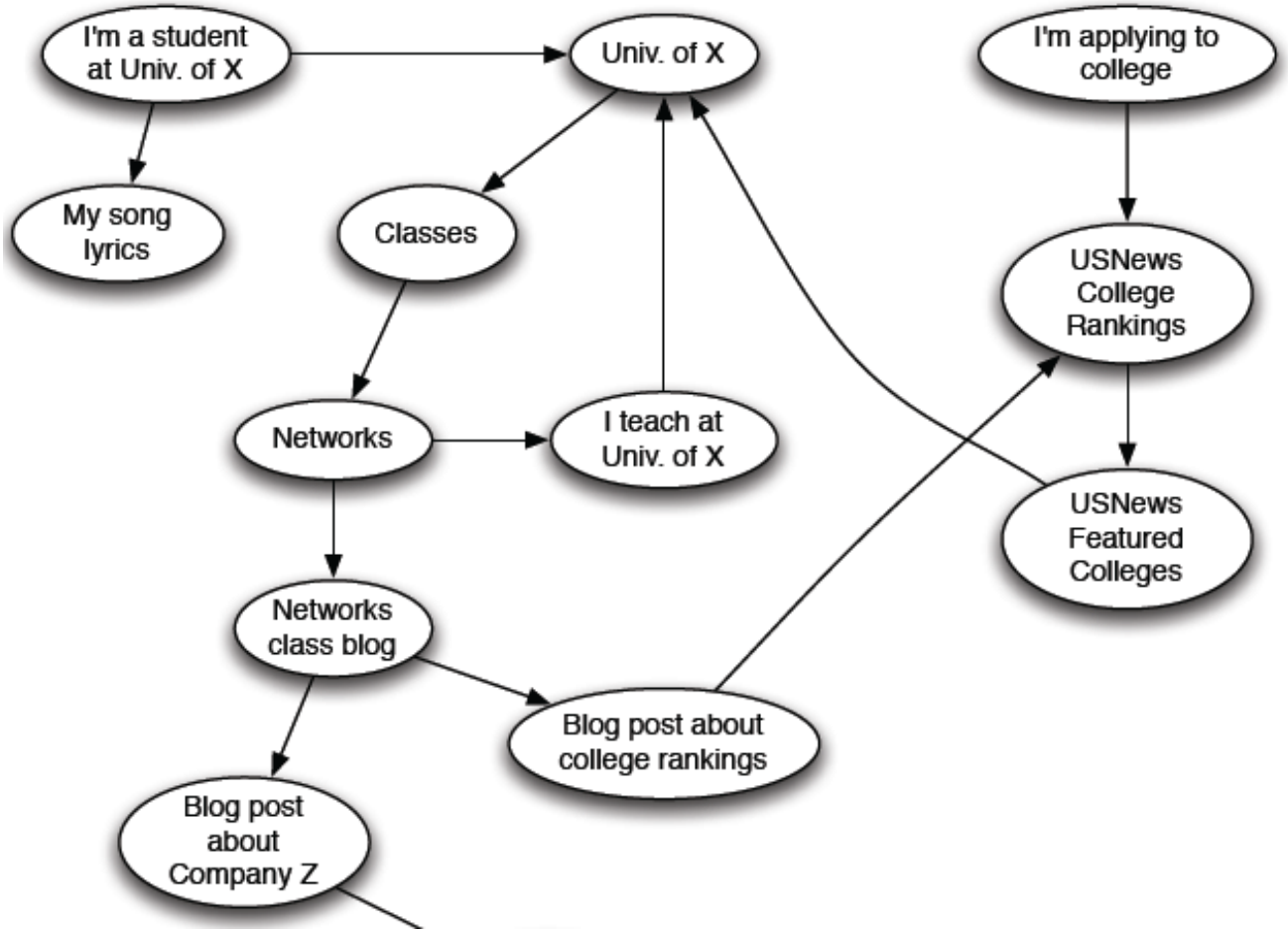
```
MATCH (p1:Person {name: 'Michael'})

MATCH (p2:Person {name: 'Karin'})

RETURN algo.linkprediction.commonNeighbors(p1, p2) AS score
```

Web as a Graph

- Web as a directed graph:
  - Nodes: Webpages
  - Edges: Hyperlinks



Web as a Directed Graph



Web as a Directed Graph

- How to organize the Web?
- First try: Human curated
  Web directories
  - Yahoo, DMOZ, LookSmart
- Second try: Web Search
  - Information Retrieval investigates:
    Find relevant docs in a small
    and trusted set
    - Newspaper articles, Patents, etc.
  - But: Web is huge, full of untrusted documents, random things, web spam, etc.

Web Search: Challenges

2 challenges of web search:

- (1) Web contains many sources of information
  Who to “trust”?
  - Trick: Trustworthy pages may point to each other!

- (2) What is the “best” answer to query “newspaper”?
  - No single right answer
  - Trick: Pages that actually know about newspapers might all be pointing to many newspapers

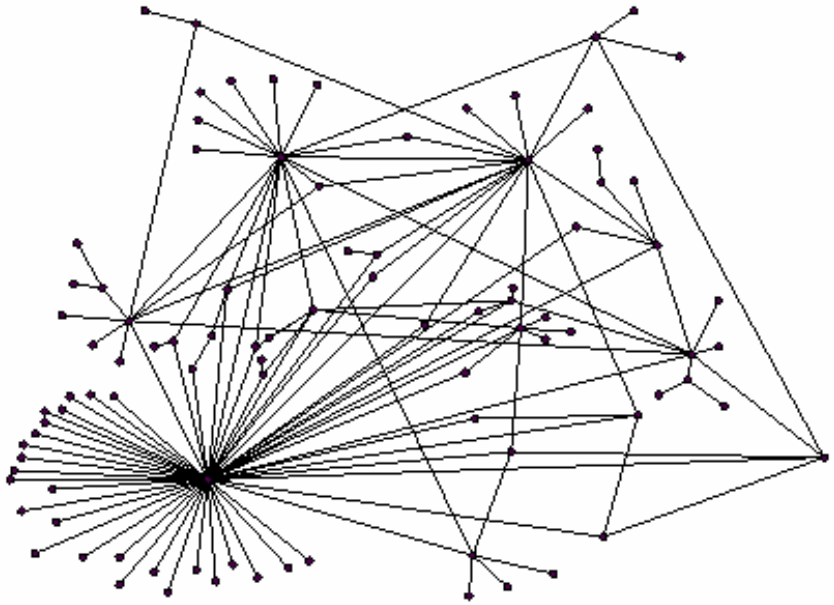
Ranking Nodes on the Graph

- All web pages are not equally “important”

  www.viteee.vit.ac.in vs. www.vit.ac.in

- There is large diversity in the web-graph node connectivity.

  Let’s rank the pages by the link structure!



Link Analysis Algorithms

- We will cover the following Link Analysis approaches for computing importances
  of nodes in a graph:
  - Page Rank
  - Topic-Specific (Personalized) Page Rank
  - Web Spam Detection Algorithms

Link Analysis Algorithms

- We will cover the following Link Analysis approaches for computing importances of nodes in a graph:
  - Page Rank
  - Topic-Specific (Personalized) Page Rank
  - Web Spam Detection Algorithms

Some Problems with PageRank

- Measures generic popularity of a page
  - Will ignore/miss topic-specific authorities
  - Solution: Topic-Specific PageRank (next)
- Uses a single measure of importance
  - Other models of importance
  - Solution: Hubs-and-Authorities
- Susceptible to Link spam
  - Artificial link topographies created in order to boost page rank
  - Solution: TrustRank

Topic-Specific PageRank

- Instead of generic popularity, can we measure popularity within a topic?
- Goal: Evaluate Web pages not just according to their popularity, but by how close they are to a particular topic, e.g. “sports” or “history”
- Allows search queries to be answered based on interests of the user
  - Example: Query “Trojan” wants different pages depending on whether you are interested in sports, history and computer security

Topic-Specific PageRank

- Random walker has a small probability of teleporting at any step
- Teleport can go to:
  - Standard PageRank: Any page with equal probability
    - To avoid dead-end and spider-trap problems
  - Topic Specific PageRank: A topic-specific set of “relevant” pages
- Idea: Bias the random walk
  - When walker teleports, she pick a page from a set *S*
  - *S* contains only pages that are relevant to the topic
    - E.g., Open Directory (DMOZ) pages for a given topic/query
  - For each teleport set *S*, we get a different vector r_S

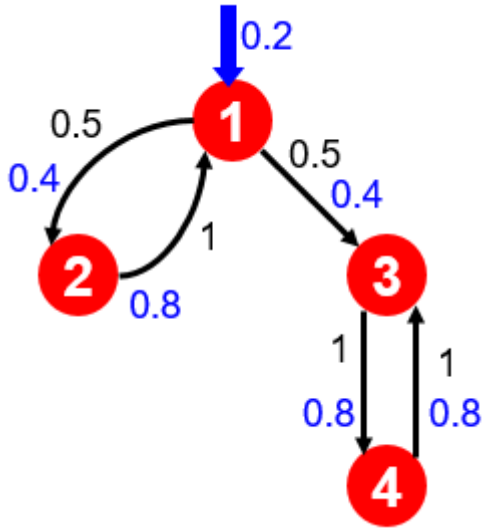
Matrix Formulation

- To make this work all we need is to update the teleportation part of the PageRank formulation:

$$A_{ij} = \begin{cases} \beta\, M_{ij} + (1-\beta)/|S| & \text{if } i \in S \\ \beta\, M_{ij} + 0 & \text{otherwise} \end{cases}$$

  - A is stochastic!
- We weighted all pages in the teleport set S equally
  - Could also assign different weights to pages!
- Compute as for regular PageRank:
  - Multiply by M, then add a vector
  - Maintains sparseness

Example: Topic-Specific PageRank



Suppose $S = \{1\}$, $\beta = 0.8$

Node	Iteration				
	0	1	2	…	stable
1	0.25	0.4	0.28		0.294
2	0.25	0.1	0.16		0.118
3	0.25	0.3	0.32		0.327
4	0.25	0.2	0.24		0.261

S={1,2,3,4}, β=0.8:
r=[0.13, 0.10, 0.39, 0.36]
S={1,2,3} , β=0.8:
r=[0.17, 0.13, 0.38, 0.30]
S={1,2} , β=0.8:
r=[0.26, 0.20, 0.29, 0.23]
S={1} , β=0.8:
r=[0.29, 0.11, 0.32, 0.26]

S={1}, β=0.90:
r=[0.17, 0.07, 0.40, 0.36]
S={1} , β=0.8:
r=[0.29, 0.11, 0.32, 0.26]
S={1}, β=0.70:
r=[0.39, 0.14, 0.27, 0.19]

Discovering the Topic Vector S

- **Create different PageRanks for different topics**
  - The 16 DMOZ top-level categories:
    - arts, business, sports,…
- **Which topic ranking to use?**
  - User can pick from a menu
  - Classify query into a topic
  - Can use the **context** of the query
    - E.g., query is launched from a web page talking about a known topic
    - History of queries e.g., “basketball” followed by “Jordan”
  - User context, e.g., user’s bookmarks, …

Summary

This session will give the knowledge about

- Link analysis algorithms
- Web as a Graph
- Topic specific page rank