**Experiment No. 10**

**Aim:** Implementation of Page rank/HITS algorithm.

**Theory:**

**Hyperlink Induced Topic Search** (HITS) is an algorithm used in link

analysis. It could discover and rank the webpages relevant for a

particular search. The idea of this algorithm originated from the fact

that an ideal website should link to other relevant sites and also

being linked by other important sites.

HITS uses hubs and authorities to define a recursive relationship

between web pages.

● Authority: A node is high-quality if many high-quality

nodes link to it

● Hub: A node is high-quality if it links to many high-quality

nodes

Algorithm Steps

● Initialize the hub and authority of each node with a value

of 1

● For each iteration, update the hub and authority of every

node in the graph

● The new authority is the **sum of the hub** of its parents

● The new hub is the **sum of the authority** of its children

● Normalize the new authority and hub

The first step that the algorithm takes is to retrieve the data of the

search query. This is the information people type on search engines

to obtain a specific result. Then, it performs a computation only regarding these results, without taking into consideration other websites.

After that, authoritative and hub values are defined, and a process

of iteration begins. In the iteration process, two updates are done:

the authority update and the hub update. For HITS, an authoritative

website is a site that has valuable content. These types of websites

tend to rank higher on the search engine results page because they

are considered ‘expert’ pages. Much like PageRank (another

algorithm that identifies and ranks sites), HITS takes the linkage of

documents on the web into account.

However, HITS differentiates itself from PageRank on a few aspects:

• It is executed at query time, not at indexing time. The hub and

authority scores assigned to a page are query-specific. This means

that the ranking will always consider the keyword or content that

people are searching for.

• Whereas algorithms like PageRank compute one score per

document, HITS compute two.

• It is processed on a small subset of documents, instead of all

documents, like PageRank does.

• Search engines do not commonly use it.

**Code:**

import numpy as np

def pagerank(M, num\_iterations, d=0.85):

N = M.shape[1]

page\_rank = np.ones(N) / N

for \_ in range(num\_iterations):

new\_page\_rank = (1 - d) / N + d \* np.dot(M, page\_rank)

if np.allclose(new\_page\_rank, page\_rank, atol=1e-6):

break

page\_rank = new\_page\_rank

return page\_rank

web\_graph = np.array([

[0, 1, 1, 0],

[1, 0, 1, 1],

[1, 0, 0, 1],

[0, 1, 1, 0]

], dtype=float)

web\_graph /= web\_graph.sum(axis=0, keepdims=True)

page\_rank\_scores = pagerank(web\_graph, num\_iterations=100)

rank\_order = np.argsort(page\_rank\_scores)[::-1]

for i, (score, rank) in enumerate(zip(page\_rank\_scores, rank\_order), start=1):

print(f"Page {i}: PageRank Score={score:.4f}, Rank={rank + 1}")

**Output:**

Page 1: PageRank Score=0.2320, Rank=2

Page 2: PageRank Score=0.3012, Rank=3

Page 3: PageRank Score=0.2347, Rank=4

Page 4: PageRank Score=0.2320, Rank=1

\*\* Process exited - Return Code: 0 \*\*

Press Enter to exit terminal

**Conclusion:** Thus, in this experiment, we have implemented Page Rank/HITS algorithm.