Discussion

In this session, we discuss the accuracy of the proposed *m-PageRank* by computing the rank of each vertex of the given Fig. 1. The algorithm is analyzed by markov chain method. Hence we find out how suitable the given *m-PageRank* recommend for multilayer networks.

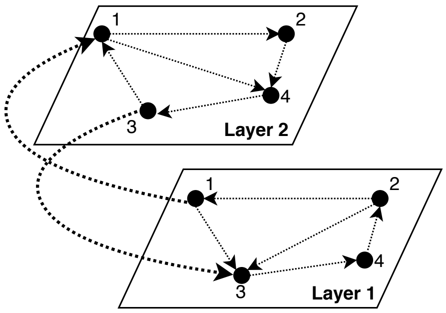

Figure 1: An illustrative example. A multilayer social network with four [1, 2, 3, and 4] nodes connected by two different relationships.

The supra-adjacency matrix represents the *column stochastic* matrix, whcih is used for computing *m-PageRank*. The supra-adjacency matrix for the given social graph is as shown below:

$$A_{Mx} = \left(\begin{array}{c|cccc|cccc}
 & 1\otimes l_1 & 2\otimes l_1 & 3\otimes l_1 & 4\otimes l_1 & 1\otimes l_2 & 2\otimes l_2 & 3\otimes l_2 & 4\otimes l_2 \\
\hline
1\otimes l_1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
2\otimes l_1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
3\otimes l_1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
4\otimes l_1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
1\otimes l_2 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
2\otimes l_2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
3\otimes l_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
4\otimes l_2 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
\hline
\end{array}\right)$$

There are four nodes in the multilayer graph, we could create an 4×4 column stochastic matrix by defining the each entry in the matrix as:

$$M = m_{ij} = \begin{cases} 1/d_o(j), & \text{if } there\ is\ a\ link\ from\ j\ to\ i \\ 0, & otherwise \\ 1, & \text{if } \langle v_i^{\alpha}, v_j^{\beta} \rangle \in E_{\alpha\beta}. \end{cases} \tag{1}$$

The distinguishable feature of the column stochastic matrix is that the sum of each column is a whole number and lies between 1 and L, where L is the number of layers in the networks.

$$M = \begin{bmatrix} 1 & \frac{1}{3} & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{2} & 0 \end{bmatrix}$$

Let us consider a column vector x of size 4×1, which consists of initial rank of each vertices.

$$x_0 = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}$$

The *m-PageRank* algorithm repeatedly replace x by the product of Mx until it converges. If there are k iterations required to converge the system, x_k is computed as:

$$x_k = M^k x_0 \tag{2}$$

$$x_k = \begin{bmatrix} 1 & \frac{1}{3} & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{2} & 0 \end{bmatrix}^k \times \begin{bmatrix} \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \\ \frac{1}{4} \end{bmatrix}$$

The column vector x_k is *m-PageRank* vector.

Table 1: The table shows the *m-PageRank*, $x_k = M^k x_0$, for each node on every iteration

Iteration	Node 1	Node 2	Node 3	Node 4
1	0.33	0.20	0.415	0.29
2	0.44	0.25	0.53	0.39
3	0.57	0.34	0.68	0.39
.. ..	.. ..	.. ..	.. ..	.. ..
7	0.74	0.43	0.88	0.64
8	0.95	0.56	1.14	0.82