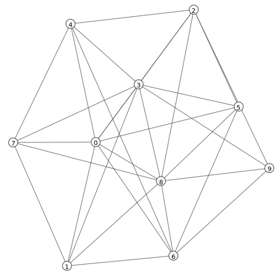

Fig. 10: A sample Erdős-Renyi graph with 10 nodes and 30 edges

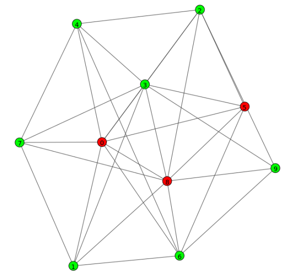

Fig. 11: Generated communities

A. An example of GloVeNoR

Here we illustrate a sample execution of the algorithm on an example graph. In Figure 10 we have a random Erdős-Renyi graph with $p = 1/2$ (note that this is different than our p in the random walk). It is an unweighted, undirected graph with 10 nodes and 30 edges. The first step of the algorithm it to generate the random walks corpus. We used the values for number of walks per node, $k = 1$ and length of a walk, $l = 5$. A sample of the output of this step is as follows: $0, 8, 2, 3, 5$

$1, 7, 4, 2, 3$
$2, 3, 5, 9, 8$
$3, 0, 8, 1, 7$
$4, 0, 7, 1, 0$
$5, 3, 2, 4, 7$
$6, 8, 0, 3, 4$
$7, 0, 8, 3, 4$
$8, 3, 0, 1, 6$
$9, 5, 2, 4, 7$

The next step generates the co-occurrence matrix using this corpus. We used a window size $w = 1$ for the above corpus. The generated co-occurrence matrix is given by:

0.0	2.0	0.0	3.0	1.0	0.0	0.0	2.0	4.0	0.0
2.0	0.0	0.0	0.0	0.0	0.0	1.0	3.0	1.0	0.0
0.0	0.0	0.0	4.0	3.0	1.0	0.0	0.0	1.0	0.0
3.0	0.0	4.0	0.0	2.0	3.0	0.0	0.0	2.0	0.0
1.0	0.0	3.0	2.0	0.0	0.0	0.0	3.0	0.0	0.0
0.0	0.0	1.0	3.0	0.0	0.0	0.0	0.0	0.0	2.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
2.0	3.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0
4.0	1.0	1.0	2.0	0.0	0.0	1.0	0.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	1.0	0.0

After obtaining the co-occurrence matrix, a random vector of d dimensions is initialized for every node. These vectors are optimized in the final step using the co-occurrence matrix and the cost function. The final step trains the GloVeNoR model using the co-occurrence matrix. This step generates d-dimensional embeddings of the graph nodes. For the above example, the dimensionality, d, of the embeddings is 2. The generated vectors are given by:

0.00032430148178421857	0.7938853234927894
0.47120730004410316	0.07747351511500729
0.6995203607385199	0.22675897204604747
0.40398774088300593	0.3402914202062996
0.3684166382730359	−0.0466443816755288
−0.09899558914305986	0.3546439090539501
0.32795573825915464	0.32323114306831036
0.3240960932604696	0.21676116252743483
−0.15893379881151024	0.7265621806582748
0.30177281552368374	0.3173293382685133

These vectors are clustered using the k-means clustering algorithm. The generated clusters are communities in the graph. In the above example, we are using a k value of 2. The generated communities are shown in Figure 11.