

Topo2vec: A novel node embedding generation based on the topology of network for the link prediction (*supplementary file*)

Koushik Mallick, Sanghamitra Bandyopadhyay, *Fellow, IEEE*, Subhasis Chakraborty, Rounaq Choudhuri and Sayan Bose



1 ALGORITHMS

In *Proposed Method* (ref: Section 3 of the main manuscript), we have described two algorithms that are used in our proposed method, Topo2vec. Here, for detailed references to the used algorithms, we have a more elaborate explanations of the same.

The Algorithm 1 (ref: Algorithm 1 of the main manuscript) performs the initial exploration of a sub-graph by mining its neighbors from the adjacency list of the concerned source node. Algorithm 2 (ref: Algorithm 2 of the main manuscript) extends the context sub-graph formation as built by Algorithm 1 by employing more extensive but stringent methodology.

Algorithm 1 To explore the initial context subgraph of each individual nodes in Graph G , $G = \{V, E\}$,

```

1: The input: Adjlist (Adjacency list data structure of
   the Graph.) Initialize  $context\_nodes \leftarrow NULL$ ,
    $node\_number = \|V\|$ .
2: for  $CurrNode \leftarrow (0 : node\_number - 1)$  do
3:    $visited[0 : node\_number - 1] \leftarrow 0$ 
4:    $visited[CurrNode] = 1$ 
5:    $templist = Adjlist[CurrNode]$ ;
6:   if  $templist == NULL$  then
7:     continue;
8:   else
9:      $size \leftarrow length(templist)$ ,  $Score[0:size-1] \leftarrow 0$ 
10:    for  $i \leftarrow (0 : size-1)$  do
11:       $Score[i] = NA_{CurrNode}^{templist(i)}$ 
12:    end for
13:     $[Score, ind] = Sort(Score)$ ;
14:     $templist \leftarrow templist[ind]$ ;
15:     $templist(i) = [\exists i, Score[i] \geq \tau]$ ;
16:  end if
17:  for  $k \leftarrow 0 : Average\_degree$  do
18:     $templist = SEARCH(templist, visited, CurrNode)$ 
19:  end for
20:   $context\_nodes = [context\_nodes; templist]$ 
21: end for

```

Algorithm 2 To expand the context sub-graph of a source node in a network.

```

0: procedure  $SEARCH(templist, visited, CurrNode)$ 
1:  $Tempauglist \leftarrow NULL$ 
2:  $Utemplist \leftarrow Uniquelist(templist)$ 
3: for  $i \leftarrow (0 : length(Utemplist) - 1)$  do
4:   if  $(visited[Utemplist[i]] == 0)$  then
5:      $visited[Utemplist[i]] \leftarrow 1$ 
6:      $neighborl = Adjlist[Utemplist[i]]$ 
7:     for  $j \leftarrow (0 : |neighborl| - 1)$  do
8:        $Score[i] \leftarrow NA_{CurrNode}^{neighborl(j)}$ 
9:     end for
10:     $[Score, indx] = Sort(Score)$ 
11:     $neighborl = neighborl[indx]$ 
12:     $neighborl(i) = [\exists i, Score[i] \geq \tau]$ 
13:     $Auglist \leftarrow neighborl$ 
14:    for  $k \leftarrow (0 : |Auglist| - 1)$  do
15:       $Score[k] \leftarrow SA_{CurrNode}^{Auglist(k)}$ 
16:    end for
17:     $[Score, indx] = Sort(Score)$ 
18:     $Auglist = Auglist[indx]$ 
19:     $Auglist(k) = [\exists k, Score[k] \geq \tau]$ 
20:     $Tempauglist \leftarrow [Tempauglist, Auglist]$ 
21:  else
22:    Continue;
23:  end if
24: end for
25:  $templist \leftarrow [templist, Tempauglist]$ 
26: return  $templist$ 
26: end procedure

```

2 ANALYTICAL ANALYSIS

The proposed algorithm Topo2vec for generation of node embeddings of a graph, has been evaluated using various operators as mentioned in *Evaluation Methods* (ref: Section 5 of the main manuscript). In *Discussion* (ref: Section 7 of the main manuscript), we have elaborated on the testing of the new feature projection method using different operators viz., Hadamard Product, SimKron_Hadamard and SimKron_Avgerage as described in Table 1 of the main manuscript. Here, we have shared the analytical analysis in

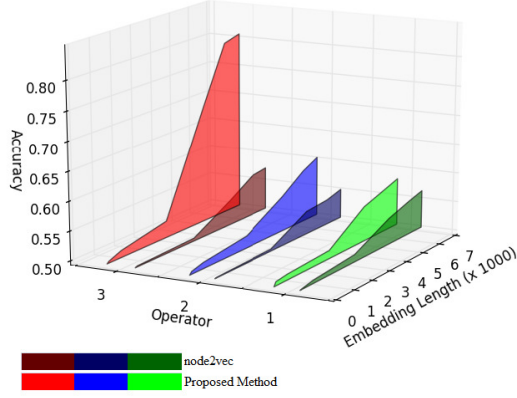


Fig. 1: Performance comparison of different pairwise feature representation operators defined in Table 1 in the manuscript, through clustering. Results are shown using PPI data (4 fold linked information) with the proposed method and node2vec. Index in the operator-axis represents (1) Hadamard, (2) *SimKron_Average* and (3) *SimKron_Hadamard* operators respectively.

the form of a 3-Dimensional plot which helps us to visualize the difference in performance among these operators.