**Project report**

Project study

Our study flow was:

* Reading the abstract and introduction part of the paper(individually)
* Discussing the purpose and the main idea of the paper(as a group)
* Read the rest of the paper (together) and while reading:
  + lookup for any term we didn't know before.
  + Discuss the term so we all understand the concept.
* Find an implementation of the DRNE algorithm in python
* Create an environment for the code to run in.
* Adjust the cond for compilation errors
* Run the code
* Discuss the results of the code and compare them to the expected result.
* Split the code into pieces, read and discuss every piece to make sure we understand each part.
* Read the code as a whole to understand the integration of each piece.

This study way helped us to make sure we understand the core of the algorithm step by step. And make sure none of the group members is left behind.

Project flow

For a given graph the DRNE algorithm

* sample the neighborhood
* sort the neighbors of each vertex by their degree
* for each vertex:
  + aggregate the embeddings of the neighbors using Layer-normalized LSTM

into the embeddings of the vertex

* + regulate the answer

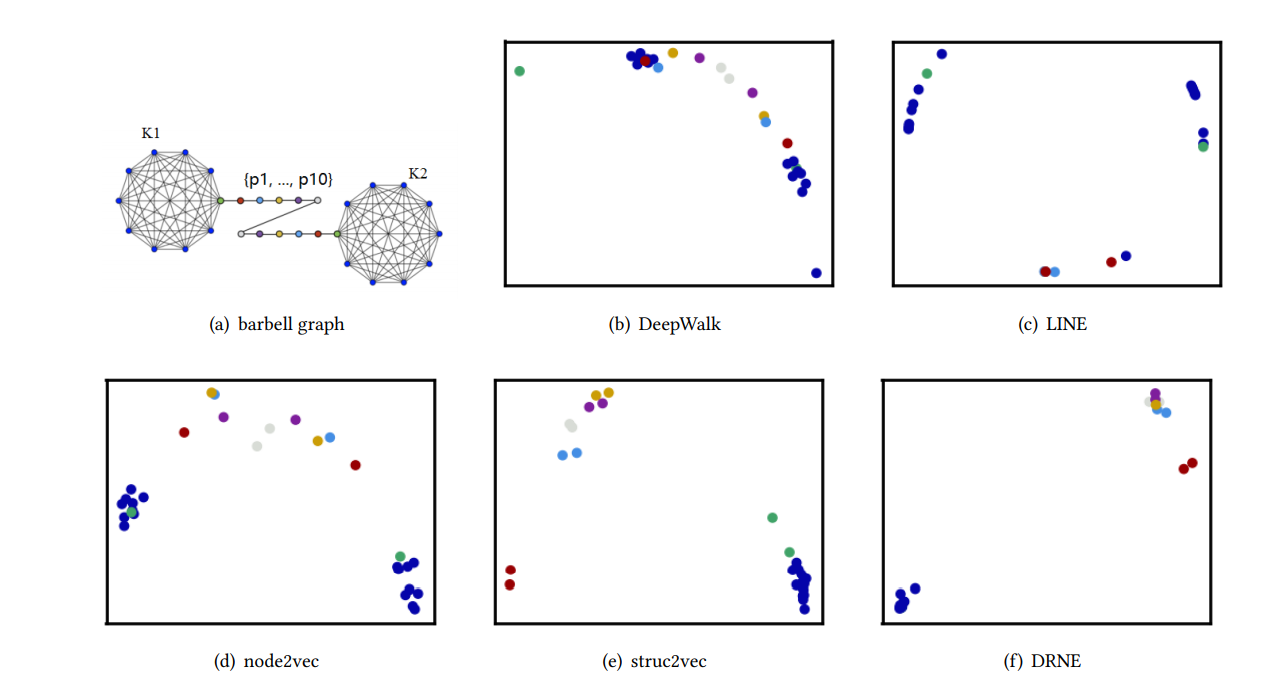
Results

Given the "barbell" dataset (shown below), our algorithm output embedding which gives us more clusterization than other algorithms,

For example in figures[(b),(c),(d)] we can see the blue vertices on the left bell (K1) and the vertices on the right bell(K2) are far from each other even though they serve the same purpose

In figure (e) these vertex groups are clustered but the pole (p1, …, p10) don’t seem to be very clustered as p1 and p10 (the red vertices) are very far away, as they connect the bell to the pole, but we can see that these vertices are more resemble the pole vertices so we will want them to be closer.

That's DRNE algorithm (figure (f)) shows the wanted results, K1 and K2 are clustered, p2, .., p9 clustered and the vertices p1 and p10 are close to the other pole vertices.



In our output the embeddings of the "barbell" dataset show the same result, as the bells vertices(K1,K2) are within the same range(0.74 – 1.00) and the pole vertices(p1, .., p10) are within the same range(0.15 – 0.27)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **K1** | **0.78** | **0.78** | **0.79** | **0.75** | **0.8** | **0.79** | **0.78** | **0.78** | **0.79** | **0.8** | **0.8** | **0.77** | **0.78** | **0.76** | **0.77** | **0.81** |
| **0.78** | **0.78** | **0.78** | **0.74** | **0.8** | **0.8** | **0.78** | **0.78** | **0.78** | **0.8** | **0.8** | **0.76** | **0.77** | **0.75** | **0.76** | **0.81** |
| **0.87** | **0.87** | **0.87** | **0.84** | **0.88** | **0.87** | **0.87** | **0.86** | **0.87** | **0.88** | **0.88** | **0.86** | **0.86** | **0.85** | **0.86** | **0.88** |
| **0.87** | **0.87** | **0.87** | **0.84** | **0.88** | **0.87** | **0.87** | **0.87** | **0.87** | **0.88** | **0.88** | **0.86** | **0.87** | **0.85** | **0.86** | **0.88** |
| **0.87** | **0.86** | **0.87** | **0.85** | **0.88** | **0.88** | **0.88** | **0.87** | **0.87** | **0.88** | **0.88** | **0.86** | **0.87** | **0.86** | **0.86** | **0.88** |
| **0.87** | **0.86** | **0.87** | **0.84** | **0.88** | **0.87** | **0.88** | **0.87** | **0.87** | **0.88** | **0.88** | **0.86** | **0.87** | **0.85** | **0.86** | **0.88** |
| **0.87** | **0.86** | **0.88** | **0.85** | **0.88** | **0.88** | **0.89** | **0.87** | **0.88** | **0.88** | **0.89** | **0.86** | **0.87** | **0.86** | **0.87** | **0.89** |
| **0.87** | **0.86** | **0.87** | **0.84** | **0.88** | **0.87** | **0.88** | **0.87** | **0.88** | **0.88** | **0.89** | **0.86** | **0.87** | **0.85** | **0.86** | **0.88** |
| **0.91** | **0.89** | **0.9** | **0.84** | **0.87** | **0.87** | **0.88** | **0.89** | **0.9** | **0.9** | **0.92** | **0.85** | **0.86** | **0.87** | **0.87** | **0.93** |
| **0.99** | **0.96** | **0.95** | **0.96** | **0.95** | **0.98** | **1.00** | **0.96** | **0.97** | **0.97** | **1.00** | **0.94** | **0.99** | **0.96** | **0.97** | **0.98** |
| **P1** | **0.22** | **0.21** | **0.23** | **0.19** | **0.24** | **0.22** | **0.23** | **0.22** | **0.22** | **0.21** | **0.24** | **0.22** | **0.23** | **0.19** | **0.20** | **0.24** |
| **P2** | **0.17** | **0.20** | **0.19** | **0.18** | **0.19** | **0.18** | **0.19** | **0.17** | **0.19** | **0.18** | **0.20** | **0.19** | **0.20** | **0.18** | **0.20** | **0.20** |
| **P3** | **0.18** | **0.18** | **0.24** | **0.22** | **0.25** | **0.2** | **0.22** | **0.21** | **0.21** | **0.18** | **0.25** | **0.24** | **0.23** | **0.2** | **0.17** | **0.24** |
| **P4** | **0.19** | **0.19** | **0.25** | **0.22** | **0.25** | **0.21** | **0.23** | **0.22** | **0.22** | **0.19** | **0.25** | **0.24** | **0.23** | **0.21** | **0.17** | **0.25** |
| **P5** | **0.18** | **0.19** | **0.24** | **0.22** | **0.25** | **0.20** | **0.22** | **0.21** | **0.22** | **0.19** | **0.25** | **0.24** | **0.23** | **0.2** | **0.17** | **0.24** |
| **P6** | **0.18** | **0.17** | **0.25** | **0.23** | **0.27** | **0.21** | **0.24** | **0.22** | **0.22** | **0.18** | **0.26** | **0.25** | **0.24** | **0.21** | **0.17** | **0.25** |
| **P7** | **0.19** | **0.17** | **0.25** | **0.23** | **0.26** | **0.21** | **0.24** | **0.22** | **0.23** | **0.18** | **0.26** | **0.25** | **0.24** | **0.21** | **0.17** | **0.25** |
| **P8** | **0.19** | **0.19** | **0.25** | **0.22** | **0.26** | **0.21** | **0.23** | **0.22** | **0.22** | **0.19** | **0.26** | **0.25** | **0.23** | **0.21** | **0.15** | **0.26** |
| **P9** | **0.18** | **0.22** | **0.21** | **0.18** | **0.21** | **0.19** | **0.20** | **0.19** | **0.20** | **0.20** | **0.21** | **0.20** | **0.20** | **0.20** | **0.17** | **0.23** |
| **P10** | **0.22** | **0.22** | **0.23** | **0.19** | **0.24** | **0.22** | **0.23** | **0.22** | **0.23** | **0.22** | **0.23** | **0.22** | **0.23** | **0.20** | **0.19** | **0.24** |
| **K2** | **1.00** | **0.98** | **0.96** | **0.94** | **0.96** | **0.97** | **1.00** | **0.96** | **0.97** | **0.97** | **1.00** | **0.94** | **0.97** | **0.97** | **0.96** | **1.00** |
| **0.87** | **0.87** | **0.87** | **0.84** | **0.88** | **0.87** | **0.87** | **0.86** | **0.87** | **0.88** | **0.88** | **0.86** | **0.87** | **0.85** | **0.86** | **0.88** |
| **0.87** | **0.86** | **0.87** | **0.84** | **0.88** | **0.88** | **0.87** | **0.87** | **0.87** | **0.88** | **0.88** | **0.86** | **0.87** | **0.85** | **0.86** | **0.88** |
| **0.87** | **0.86** | **0.87** | **0.84** | **0.88** | **0.88** | **0.87** | **0.87** | **0.87** | **0.88** | **0.89** | **0.86** | **0.87** | **0.85** | **0.86** | **0.88** |
| **0.87** | **0.87** | **0.87** | **0.85** | **0.88** | **0.88** | **0.88** | **0.87** | **0.88** | **0.88** | **0.89** | **0.86** | **0.87** | **0.85** | **0.86** | **0.89** |
| **0.87** | **0.86** | **0.87** | **0.84** | **0.88** | **0.88** | **0.88** | **0.87** | **0.87** | **0.88** | **0.89** | **0.86** | **0.87** | **0.85** | **0.86** | **0.88** |
| **0.87** | **0.86** | **0.87** | **0.84** | **0.87** | **0.87** | **0.88** | **0.87** | **0.88** | **0.88** | **0.88** | **0.86** | **0.87** | **0.85** | **0.86** | **0.88** |
| **0.87** | **0.86** | **0.88** | **0.85** | **0.89** | **0.88** | **0.89** | **0.87** | **0.88** | **0.88** | **0.89** | **0.86** | **0.87** | **0.86** | **0.87** | **0.89** |
| **0.86** | **0.86** | **0.88** | **0.85** | **0.89** | **0.87** | **0.89** | **0.86** | **0.88** | **0.88** | **0.90** | **0.86** | **0.86** | **0.85** | **0.86** | **0.89** |
| **0.91** | **0.89** | **0.91** | **0.85** | **0.87** | **0.87** | **0.89** | **0.89** | **0.91** | **0.91** | **0.93** | **0.86** | **0.87** | **0.88** | **0.87** | **0.93** |

Conclusions

The DRNE algorithm gives better results for regular equivalence than the other algorithms we compared( deep walk, LINE, node2vec, struct2vec).