Node2Vec paper Report

Name : Mark Callan

Student Number : 02214113

Problems and contributions

The authors propose Node2Vec a semi supervised learning algorithm for scalable feature learning in networks , the algorithm aims to address the previous issues of feature learning being tedious and time consuming and not scalable. The paper proposes defining a flexible definition of what a network neighbourhood is , efficiently optimizing a novel network-aware neighbourhood preserving objective function using stochastic gradient descent.

Methodology

Node2vec learns node embeddings by simulating 2nd-order random walks, controlled by p and q receptively to balance local (BFS) and global (DFS) neighbourhoods. It optimizes a Skip-gram objective via SGD with negative sampling, generalizing DeepWalk and LINE. The paper proves walk convergence to graph properties and shows empirical gains (e.g., 27% better multi class classification , 12% better link predictions) on BlogCatalog and PPI. Its flexibility in defining neighbourhoods is significant and is one of the core differentiators of Node2vec

Implementation and Insights

For the Facebook dataset. I implemented node2vec using the parameters : dimensions=128, walk_length=80, and num_walks=10 to capture community structures, setting p=0.25 and q=0.25 to emphasize homophily which is valid for our social network dataset and window=10 for more precise similarity's. I used dictionary comprehensions for embedding generation and feature concatenation as I wanted to minimize the runtime for thousands of nodes. I deviated from Node2Vec by concatenating normalized feature attributes (feat.txt) , The goal of this was to enhance the embeddings with the profile data from feat.txt. Our results in similarity_results.txt range from 0.1845 (1410 - 1385) to 0.9522 (1601 – 1880) with a mean of 0.6233.

These results indicate that tight knit friendship groups are amplified by shared attributes e.g. this shared attribute may be following the same celebrity. High values such as those >0.9 indicate feature dominance which may risk overfitting , conversely low values indicate distant nodes. The extremely high score for 1601 – 901: 0.9501 is interesting

because the high similarity score is not because of shared connections in edges.txt but they are also similar because of shared features e.g. the two users may share the same birthday or hobby. Our first result 1043 - 1043: 1.0000 is as expected as the cosine similarity of comparing the same vector to itself should be 1.0 indicating that the code is correctly computing similarity

Limitations

- **Reliance on P and Q :** Node2Vec is heavily reliant on the values of P and Q are requires tuning to find the optimal value for a given graph , This value may not generalize well to different networks such as undirected graphs
- **Exclusion of features :** The original algorithm only uses graph structure and ignores the attributes of nodes which can lead to semantic data (such as user interests or birthdays in our case) not being factored in. We can see in our results that for pair : *1601 - 901: 0.9501* factoring in these features can add value
- **Scalability for huge graphs:** Although the running time of (O(|V|·K) per walk is quite good when we apply this to huge graphs for example the full graph of Facebook and not just a small sample like we use , we are bound to run into memory and time complexity issues unless tweaked further