Name: Prudhviraj Sheela | OSU ID: A20228857

Name: Aman Masipeddi | OSU ID: A20198116

CS 5423 Final Project Report

Core2Vec: A core preserving feature learning framework for networks

(Authors: Soumya Sarkar, Aditya Bhagawat, Animesh Mukherjee)

(Group Name: Data Techies)

1. Research Problem

The project that we have done is of an implementation flavour project. In this project we had tried to understand and implement the Core2Vec algorithm with additional features to it. The research problem states that for a given graph how can we obtain a framework algorithm that can help in categorizing the graph data i.e., the nodes and the edges that are arranged based on the factors of closeness or separability within the network which is also called as the semantic relatedness of the data. The Core2Vec is such an algorithm for learning low dimensional continuous feature mapping of data. The embedding network obtained is a core periphery structure where the core nodes are interconnected densely and the peripheral nodes are adjacent to the core nodes. These two disjoint layers have separate functional and structural properties of nodes within the network.

In this project we have used the algorithm mentioned by the author in the proposed paper and have applied required modifications to identify different factors within the embedding network. The initial analysis that we have performed using the existing algorithm is for the given data input based on the source node chosen which is the initial core in the network we assign the neighbouring nodes to this selected core node based on the calculated transition probability of the neighbouring node. If the neighbouring node value is greater than '0' then it implies that the node is included in the source node's core else the neighbouring node initiates as a second core node in the network which becomes the source core node of that particular core where this procedure continues until we assign the transition probabilities to all the nodes in the network and form the resultant cores in the network.

Once the periphery core structure is obtained in the embedding network, we evaluate the strength of individual cores by assigning a ranking distribution method to them using the Kendall's Ranking Correlation Coefficient method. So, on obtaining the ranking values we can decide based on the given input which core has the maximum strength where all the nodes of that particular core are strongly connected.

2. Literature Survey

There are large number of advancements being carried out in the research related to network embeddings. The related previous researches that were in the area of identifying the word co-occurrences within a network, identifying the value of word associations based on whether the words classified in the model are external objects of a word consisting of all utterances made in the speech community or how the relation of words is resided in the brains of the speakers**.**

The different other previous research categories include how to perform context-based search for automating keyword search operations using the natural language processing techniques using the selected text by the user over the web, methodologies to evaluate the time complexity of obtaining cores in a network and other relevant scalable feature learning of networks.

The different previous works that were carried out in the fields related to semantic relatedness are obtaining word co-occurrence networks using the spectral analysis which is a technique that helps in revealing the global structural patterns which shows interacting entities even in a complex network which is obtained from the set of various eigen values and their related multiplicities. The other models that were developed using semantic relatedness are PMI^{+} model for semantic structures which apprehends the semantic information in a raw word association data and does not apprehend the core structures associated with it.

The Context Based Search System is one of the previous models where the context of the given query is considered as an ordered input for finding the semantically related data of the query. This technique provides a more precise outputs when a system captures for the surrounding text. Some of the augmentation methods that use this approach are Heuristic Query Augmentation System and Linguistic CBS System. Some of the other previous feature learning algorithms developed are Node2Vec, DeepWalk and Line which extends the properties of Skip Gram Model and maximize the objective function which maximizes the log probability for observing the network neighbourhood. The optimization problem can be tractable by considering assumptions such as Conditional Independence and Symmetry in feature space.

The neighbourhoods within the network are not just limited to immediate neighbours in the structure but can vary vastly by structures depending on their sampling strategy. There are different classic search sampling strategies that are used to solve this problem. These sampling techniques are Breadth-First Sampling (BFS) and Depth-First Sampling (DFS) which represent dimensions in the term of search space they explore on learned representations. These two samplings play an important role in producing representations that reflect either the homophily hypothesis or structural equivalence.

On drawing out observations from the above models we can check out that the Core2Vec draws some of the properties from these models in the view of semantic relatedness in the embedded network. The core2vec is a more advanced framework extending the above models which is used in evaluating how the strong word associations are obtained based on the coreness of the data that is represented on top of the network structure and it also uses an effective objective functions compared to the other feature learning frameworks.

3. Methodology

I. Algorithm

The core2vec involves three stages of implementation process for obtaining the resultant graph on the embedded network. They are explained as follows:

a. **Obtaining K-Cores:** This is the initial stage of the algorithm where for the given data set or graph data, we obtain the core periphery structure by partitioning the nodes in the dataset based on their neighbouring nodes attached.

procedure KCore
Input: Graph (G) = (V, E)

Output: C[K], K = |V|

k ← 1

while |V| ≥ do

    while true do

        remove all vertices with degree ≤ k

        until all remaining vertices have degree ≥ k

        ∀ vertex(v) removed, C[v] ← k

    K ← k+1

Return (C)

b. <u>Learning Features of the data and obtaining Kendall Rank Coefficient for each core</u>

In the second phase of the algorithm after obtaining the core periphery structure we compute the pre-processing transition probabilities for all the nodes in the data set for identifying which node belongs to which core by generating random walks. Once the pre-processing transition probabilities are computed using the Stochastic Gradient Descent we place all those values into the probability transition matrix and based on the values obtained for each core of the core periphery structure, the algorithm decides which values are present in the core and which are of the peripheral. Then the Kendall Ranking Correlation Coefficient is found for individual core in order to identify which core has the maximum strength in the embedded network.

procedure LearnFeatures(G = (V,E,W), dimensions d, walk per node L, walk length l, context size c, exploration parameter p, q and penalty parameter (D), Probability Transition Matrix (P))

    core_dictionary ← KCORE(G)
    P ← PreprocessProbability(G,p,q,D,core_dictionary)
    G' ← (V,E,P)
    for all nodes u ∈ v
        Initialize walks to empty
        Walks ← genWalks (G',u,l,L)
    f ← SGD(c, d, walks)
    return f
    for all cores ∈ G
        Kendall_Rank ← Internal strength of individual core
    return Kendall_Rank

c. <u>Generating Random Walks</u>

This step is an internal part of learning features of data. This algorithm helps in

generating random walks based on the given source node and identifies the surrounding neighbours within the walk length and the total walk required. Based on this inputs we obtain the path of the walk.

```
procedure genWalks
    Input: G' (V, E, P), start node (u), walk length(l), total walk(L)
    Output: walk
    Assign walk ← u
    Assign walks ← {}
    for num_walks = 1 to L do
        for walk_iter = 1 to l do
            curr = walk [-1]
            N_curr ← Neighbours_Set(curr,G)
            S ← Sample(N_curr, P)
            walk ← s
        walks ← walk
    return walks
```

II. Analysis and Implementation of Core2Vec

For analysing the core2vec model initially we have done some research on different research papers which were based on the word association networks and semantic relatedness factors. Also there are similar feature learning frameworks such as Word2Vec, Node2Vec, DeepWalk, Line, PCM^+ model which helped in understanding the concept of semantic relatedness in the embedded networks.

The analysis of core2vec model has been developed by overcoming the limitations of random walk based techniques as it was not possible to form complex networks such as technological, semantic and biological networks which are highly disassortative. In accordance to this drawback the core2vec model was taken into consideration by the authors. The Core2Vec helped in analysing different data sets based on their semantic relatedness and process them in the embedding network as core periphery structures.

The implementation of core2vec involves the steps that are mentioned in the above algorithm. So, accordingly for implementing the core2vec the initial step is to analyse the given data set by categorizing them into a core periphery structure in the embedded network. The core nodes contain vertices which are similar and are strongly connected in the core. The peripheral nodes are adjacent to the core nodes but not similar. Then obtaining the core periphery structure we evaluate the objective function which is the average likelihood value which is to be maximized.

For the given set of vertices C_v the requirement is to maximize the probability $P(C_v \mid v)$ where the C_v consists the context vertices of 'v'. The objective function is formulated for the optimization problem as,

$$\underset{max}{f} = \sum_{v \in V} P(\mathbb{C}_v|v) = \sum_{v \in V} \prod_{\mathbb{C}_{v_i} \in \mathbb{C}_v} P(\mathbb{C}_{v_i}|v)$$

In the above formula, (C_{vi}) denotes the context node belonging to the context set (v). The value of P (C_{vi} | v) is computed using the equation,

$$P(\mathbb{C}_{v_i}|v) = \frac{\exp(v^w.\mathbb{C}^w_{v_i})}{\sum_{v\in V}\exp(v^w.\mathbb{C}^w_{v_i})}$$

In the above formula the v^w, C_{vi} represent the vector notion of node 'v' and its context node C_{vi}. The process of generating context nodes for individual source node is done by evaluating (L) random walks which are of fixed length (l) with the source node as the initial vertex. The same core nodes are aligned adjacent to adjacent to each other or are separated based on distant hops. This evaluation of context nodes with respect to a particular source node is done either using a depth-first sampling or breadth-first sampling.

Once the objective function and the necessary context nodes are analysed based on the source node, then the further methodology for finding the transition probability is by considering a random surfer where the source node is 'i' and is currently at node 'j' and it is not essential that (i,j) ∈ E (Set of edges in the input data set). The conclusion for moving to the next destination 'k' in the random walk for the edge (j,k) ∈ E is given by,

$$P_{i,j,k} = (\pi_{i,j,k} * w_{j,k}) / Z$$

In the above formula $P_{i,j,k}$ denotes the probability of the random walk which starts at vertex 'i', and is at vertex 'j' at present to move to the next adjacent destination 'k', $\pi_{i,j,k}$ is the un-normalized transition probability w.r.t the destination node, $W_{j,k}$ denotes the weight of the edge (j,k) and Z is the normalization constant. The default value of an edge for an unweighted graph is '1'. The un-normalized transition probability is computed using the formula,

$$\pi_{i,j,k} = \begin{cases} \frac{1}{(|c_i - c_j|+1)*\mathcal{D}*\lambda} & (j,k)\in E, i = k; \\ 1 & (j,k)\in E, i \neq k; \\ \frac{1}{(|c_i - c_j|+1)*\mathcal{D}*\gamma} & (j,k)\in E, i \neq k, (k,i)\notin E \\ 0, & \text{otherwise} \end{cases}$$

In the above formula, c_j, c_k represents the core id for the vertices j and k. The 'λ' and 'γ' are varied accordingly to obtain information of the close neighbourhoods of source node or distant neighbourhoods of the start node.

On calculating the above parameters, we obtain the necessary information of individual cores based on the internal transitional probabilities of each other nodes attached to the source node in that particular core. Then as an additional extension to the core2vec algorithm we have added a ranking methodology called the Kendall ranking correlation coefficient method which measures the strength of the internal nodes of the individual core. This helps in analysing from the obtained network which core has the maximum strength in the terms of closeness. In genera the mathematical formula for Kendall's ranking correlation coefficient is given by,

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{\binom{n}{2}}$$

In accordance to the formula the no. of concordant pairs is the no. of nodes within the core and the no. of discordant pairs is the no. of peripheral cores in the network. The denominator is the number of ways to select the adjacent nodes within the core, which is equivalent to [n*(n-1) / 2].

4. Evaluation and Results

The core2vec algorithm explained in the above methodology is evaluated using different data sets which are based on semantic relatedness. The data sets are collected from the University of South Florida Word Association Project and Small World of Word Project (SWOW) where the data is being trained. In this particular word association projects huge number of people had given ranks for the words based on their semantic relatedness for individual data set and associated scores between the pair of words. These scored data sets are considered as input to the core2vec algorithm for evaluation process.

The input datasets used for the testing purpose are 'Simlex', 'WordSim1', 'WordSim2', 'Mturkt' which are based on word associations and 'lesm' is an input dataset which is based on number association of data. As part of the input we also specify the filename of the output, the dimensional space, the walk length, the total number of walks involved, penalty parameter (D) which is used to reduce random runs of high core difference, proximity parameter (λ or p), distant hop size (γ or q), window size, no. of iterations, whether the given input data set is weighted or unweighted and whether the given input data set is directed or undirected.

Based on the above data inputs we obtain our core periphery structure with the resultant transition probabilities of nodes involved in the particular core and also we compute the total time required for forming the embedding network. Based on the output file obtained for the input data set which contains information about the cores obtained the kendall's ranking correlation coefficient is calculated for each core and the resultant values are displayed as output. So, on comparing all the core ranking values we can conclude that which core has the highest strength with its internal nodes.

On following the above procedure for our considered input datasets and its additional parameters different results were observed. For the number associations input data set 'lesm' the highest ranked core is 'core-0' with a value of 0.538 amongst the 36 other core obtained and the total time taken to obtain the network is 0.2423 seconds. Next, we had computed for the word association dataset 'wordsim1' the highest ranked core is core-14 with a value of 0.642 amongst the 65 other core nodes obtained and the total time taken to obtain the network is 0.4438 seconds. In the same manner we had computed for other input datasets amongst which the numbers associated data has better efficiency and the number of cores obtained are less which indicates there exists high closeness ratio for the data points in the data set.

The use of word associations data takes more computation time as it needs to process individual word based on their similarity and then create an edge with the source node of a particular core. The result output files obtained with the cores and the associated probability transition matrix gives a detail note of which vertices of positive value (>0) that indicates closeness factor of nodes that are connected to that particular core and which nodes are not connected to that particular core of negative value (<0) that indicates the separability factor. These were the results observed on running the core2vec algorithm for the considered data sets.

5. Conclusion

This implementation project has helped us in understanding the network embedding tasks using different semantic related data to identify the coreness of a node and the ranks associated with individual cores obtained. This approach firstly helped us in analysing how the given data helps in bringing the similar core nodes in close proximity and moves apart the different peripherals which are not of the close proximity based on the probability transition matrix obtained for individual core.

The approach of assigning ranks for individual core using the Kendall's ranking correlation coefficient method helped us in finding out for a given input data set from the number of core obtained we could analyse which core has the maximum strength associated within with the internal nodes of that particular core. This approach applied is an extension to the existing core2vec model and does not involve any duplication of the proposed author's work.

6. Contributions and Duties in the Project

In the initial phase of selecting this project my partner (Aman) and I (Prudhvi) have decided to do a project related to network embedding and its classification. Based on this we have chosen core2vec technique as our implementation flavour project. We both have referred different papers for doing the relevant literature survey related to this topic which include papers on word associations, semantic relatedness and their associated techniques.

Later my partner has gathered relevant data sets from the different word association groups SWOW and USF projects. Then I had decided to use these data sets collected for testing of the developed algorithm. As part of the method needed for finding ranking correlation coefficient we both decided to use Kendall's technique as this could be used more precisely evaluate rank for each core obtained. I have inherited the existing algorithm and programmed accordingly to our final goals.

My partner has done the testing part related to the algorithm developed and observed different cases obtained in the data sets. From the results obtained I analysed the conclusion that is obtained from our analysis. We both in accordance have prepared the final project report based on our requirements considered and have arranged them accordingly which state the entire process of the core2vec technique. This project on a whole helped us both in understanding how important are the network embedding structures and its techniques for analysing large semantic related data sets.

7. Source Code Files and Input Data Sets

Below is an attached GitHub hyperlink that consists of all the source code files, the input data sets, the output files obtained for the input data sets, the screenshot of the command prompt containing the output of kendall's ranking technique and a readme file which consists of the information of the software used, the libraries used, the commands used to execute the program.

GitHub repository link: https://github.com/prudhvi193/DataTechies-Core2Vec-.git