Project Report
Project: COMMUNITY DETECTION
Unity Id: ndgandh2
Paper 3 - Large-Scale Spectral Clustering on Graphs

Introduction: This document contains the implementation of Efficient Spectral Clustering on Graphs with Regeneration(ESCG-R) for large scale graph data for community detection. The algorithm aims to mitigate the computational bottleneck by reducing the size of the graph, while still providing high-quality clustering results, as compared to standard spectral methods. The report describes the nature and type of graph used for this algorithm followed by the community definition and scoring function to evaluate the crispness of the communities. Further, the ESCG-R algorithm is described in brief with its parameters and constraints. Lastly, the performance of the algorithm and the empirical results on different graph datasets with respect to goodness metrics are listed.

Graph Description: The community detection is performed on the graph data-set of amazon, youtube and dblp data. These data files contain the vertex id pair separated by the new line which is parsed and loaded into graph-like data structures for algorithm implementation. The graphs considered for these algorithm are **Undirected, Weighted and Dense.**

Community Definition:
Community can be defines as a densely connected subgraphs such that nodes within the same cluster have more connections than those in different clusters. Graph considered for the ESCG-R algorithm are well separated, non overlapping and dense.

Assumed Notations in the algorithm:

- Undirected weighted graph G = (V, E)
- Num. nodes n = |V|
- Num. Edges m = |E|
- Adjacency matrix W = $\{W_{i,j}\}_{i,j = 1,2,..,n}$

Objective function:

$$\min_X T_r(X^T D^{-1/2} L D^{-1/2} X) \text{ s.t. } X^T X = I$$

where, $X \varepsilon \Re^{n*k}$ is a matrix consisting of the column vectors, k is the number of clusters, L is graph Laplacian of W, D = L− W

This function is solved by eigenvalue decomposition (EVD).

Constraints and Relationships:
All communities detected using Efficient Spectral Clustering on Graphs with Regeneration algorithm are mutually exclusive i.e. non-overlapping disjoint communities.

$$\mathbf{R(S_i, S_j) = S_i \cap S_j = \phi}$$

where, $R(S_i, S_j)$ defines the relationship between the communities

Algorithm:

Project Report
Project: COMMUNITY DETECTION
Unity Id: ndgandh2
Paper 3 - Large-Scale Spectral Clustering on Graphs

Two alternative approaches are proposed in the paper for large scale graph clustering. The first approach used one-step generation of supernodes and second approach uses iterative method to regenerate supernodes based on the clustering result obtained from the first algorithm. Algorithm starts with generation of meaningful supernodes which are connected to the original graph. The shortest path algorithm is applied to create disjoint subsets of original graph nodes, where nodes correspond to the nearest supernode. This helps compress a large scale graph. Next, Bipartite Graph is generated which preserves the links between original graph nodes and new supernodes and is computationally efficient for spectral graph clustering. Algorithm assumes that supernodes may behave as a measure for clustering of nodes in the original graph. Moreover to reduce more computational cost the spectral clustering is carried out using singular value decomposition of the generated bipartite graph.

Following are the approaches in detail:
Algorithm 1: Efficient Spectral Clustering on graphs (ESCG)

- Input: Adjacency matrix W for graph G, number of cluster k, desired number of supernodes d
- Output : Clustering of nodes in G (Set of Communities)
- Description: Algorithm selects d random seeds and computes shortest path to all other nodes in the graph. Then nodes are partitioned into d disjoint subset with shortest paths to supernodes. Generate d supernodes and connect them to d disjoint subsets. This generates a binary matrix R. Then we compute the adjacency matrix of the transformed bipartite graph. Then eigenvalue decomposition (EVD) is converted to Singular value decomposition (SVD) to reduce computational complexity. Then from SVD, left singular vector retains information about supernodes while right singular vectors contains information about the original nodes. Top k eigen vectors of right singular vectors are stored in matrix U. Then K-means algorithm is applied on matrix U for clustering.

Algorithm 2: Regeneration of Supernodes

- Input: Embedding Matrix U
- Output: Sparse Matrix R describing links between supernodes and original nodes
- Description: Embedding matrix U indicates the partition of nodes in the graph. So the supernodes are adjusted and connected to nodes in the graph. Here the supernodes are reduced from d to 2k-2.

Algorithm 3: Efficient Spectral Clustering on Graphs with Regeneration (ESCG-R)

- Input: Adjacency matrix W for graph G, number of cluster k, desired number of supernodes d, number of iterations t
- Output : Clustering of nodes in G (Set of Communities)
- Description: In this algorithm, for the first iteration Algorithm 1 is called to generate an R binary matrix. Then SVD is performed to generate a U matrix containing the information of the original nodes. Then for the next (t-1) iterations R is computed using Algorithm 2.

After t iteration, K-means clustering is applied on final U matrix to generate clusters which denotes the output communities.

Performance Metrics:

1. **Parameterized:**
   - ESCG-R is parameterized and has 3 input parameters namely A) Number of Supernodes B) Number of iterations for supernode regeneration C) Number of communities.
   - Performance measures for the algorithm ESCG-R when run on small graphs of Amazon, Youtube and Dblp with various different values of parameters are shown below:

Graph	Supernodes (d)	Iterations	Communities (k)	Precision	Recall	Runtime (in secs)
amazon.graph.small	1000	10	216	0.759600	0.7757	243.90
youtube.graph.small	1800	3	866	0.003582	0.2794	174.00
dblp.graph.small	1200	10	311	0.723148	0.7831	351.54

   - Most critical parameter on which the algorithm depends is the number of supernodes. Supernode generation and selection is random, so the probability of getting a supernode in each community for higher precision depends on the number of supernodes.
   - Also number of iterations account for the convergence of the right set of supernodes. So number of iterations should be sufficient enough to achieve right set of supernodes.
   - Amazon and DBLP graphs are dense, so the ESCG-R algorithm performs better but, Youtube graph is very sparse(avg. ~3 nodes in a community), so ESCG-R performs poorly.

2. **Time Complexity:**
   Algorithm 1 : (ESCG) -> $O(nd \log n + md + nd^2)$
   Algorithm 2 : (Regeneration of supernodes) -> $O(mk)$
   Algorithm 3 : (ESCG-R) -> $O(nd \log n + md + mkt + n(d^2 + k^2t))$

3. **Iterative:**
   - ESCG-R is an iterative method which regenerates meaningful supernodes with guarantee of convergence after t iterations.

Project Report
Project: COMMUNITY DETECTION
Unity Id: ndgandh2
Paper 3 - Large-Scale Spectral Clustering on Graphs

4. **Randomized:**
   - ESCG-R depends critically on number of supernodes and its random generation, so the output communities are not deterministic.

5. **Not Parallelizable:**
   - As each regeneration of supernodes depends on the previous set of supernodes, program cannot be parallelized.

Goodness Metrics :

Goodness Metrics for the test graphs are as follow:

Graph	Communities	Separability	Density	Cohesiveness	Clustering coefficient
amazon.graph.small	216	23.8744370	0.494776	0.2830175	0.5485326
youtube.graph.small	866	0.06506267	0.791623	0.7888760	0.0170323
dblp.graph.small	311	18.2840444	0.771281	0.6156803	0.8371713

REFERENCES

[1] http://www.cs.cmu.edu/~xinleic/papers/aaai11.pdf
[2] http://cs.stanford.edu/people/jure/pubs/comscore-icdm12.pdf
[3]http://www.slideshare.net/akisatokimura/paper-review-largescale-spectral-clusterig-on-graphs
[4] http://en.wikipedia.org/wiki/Singular_value_decomposition