Project 2 : Community Detection
UnityId : akagrawa
Paper No : 4
Publication : Towards Linear Time Overlapping Community Detection in Social Networks

Introduction: This document contains the description of Speaker-Listener Label Propagation Algorithm (SLPA) for community detection. The algorithm detects the overlapping community in the graph-based dataset. 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 SLPA is described in brief with its parameters and constraints. Lastly, the performance of SLPA and the empirical results on different graph datasets with respect to goodness metrics are listed. Please refer **slpa.R** for the algorithm implementation code and **README.md** file for the code description and usage. For the output community results, please refer the folder "**output communities**".

Graph Description: The community detection is performed on the graph data-set of **amazon, youtube and dblp** data. These data files contain the vertex-vertex connection separated by the new line which is parsed and loaded into graph-like data structures for algorithm implementation. From the given dataset, we can conclude that the graphs are **simple(no parallel and self-loop), static, undirected, unweighted and unlabeled**. For small and medium versions of data-sets are **sparse** but the large and original versions of graphs are **dense**.

Algorithm: "**Speaker-Listener Label Propagation Algorithm"**
In this algorithm we detect the overlapping communities by the model which mimics the way the communication happens between speaker and listener. Listener listens to its speakers one at a time and preserves the most popular opinion. Here every node will be a listener and its neighbours will be speakers. It is an extension of Label Propagation Algorithm[2] which assigns the label to the nodes of the graph based on the probabilistic frequency of occurrence of its neighbour labels. Here instead of assigning just one label to the node, the node maintains a memory in which it preserves all the nodes assign to it which is reduced in the post-processing step based on threshold.

The algorithm is implemented in three stages:
Stage 1 : Initialization stage
In this stage every node is initialized with the label equal to its own vertex-id.

Stage 2 : Evolution stage
In this stage following steps are repeated until the stopping criterion is satisfied:

a) A node is selected as a **listener** from the pool of vertices of the graph in random order.
b) Now as a **speaker** each neighbour of the selected node sends a label with respect to a speaking rule i.e. selecting a random label from its memory with probability proportional to the occurrence frequency of this label in the memory.
c) The listener received the label sent to it by its speaker nodes and accepts the label with respect to a listening rule i.e. selecting a label which is having most frequency which will mimic as a most popular label.

Stage 3: Post-processing stage
In this stage the less probable labels are deleted which is having probability distribution less than certain threshold r. Then nodes having same labels are grouped together to form a community. In this way many nodes can be a part of multiple communities as **overlapping nodes**.

Community Definition and Scoring Function: A community in the graph data set will be a set of connected nodes having high separability, more cohesiveness and less conductance value.
Scoring function is evaluated as

Scoring function f(s) = **conductance** = $\frac{cs}{2ms + cs}$

where cs = number of edges on the boundary of community s that point outside of s.
ms = number of edges in the community s

Other scoring functions as **density, clustering coefficient, FalkeODF, cut-ratio, cohesiveness, separability, triangle participation ratio (TPR), fraction over median degree (FOMD)** are also evaluated for the detected communities.

Constraints and Relationships: SLPA algorithm aims to find the communities which are overlapping as well as it detects the overlapping nodes which belongs to more than one community.

a) $R(S_{i,} S_j) = S_{i,} \cap S_j \neq \Phi$, the communities are overlapping

b) $\exists u \in (S_{i,} \cup S_j)$, there exist some node u which belongs to more than one community.

Performance Metrics: The performance of the slpa is listed with respect to the experiments conducted for the amazon, youtube and dblp medium and small graphs.

	Algorithm Parameters	Algorithm Parameters	Performance Measure	Performance Measure	Performance Measure
Graph	**Threshold r**	**Iterations**	**Recall**	**Precision**	**F-measure**
Amazon Small	0.0006	20	0.8545842	1.0	0.92159116
Amazon Medium	0.0006	20	0.64757952	0.99941765	0.78591805
Youtube Small	0.0004	20	0.95469108	0.87463312	0.91291028
Youtube Medium	0.000097	20	0.96468961	0.86518127	0.91222981
Dblp Small	0.0007	20	0.81622365	0.98999444	0.89475006
Dblp Medium	0.0005	20	0.79763836	0.95913722	0.87096457

In the above table, the graphs performance metrics have been listed based on algorithm parameters **r** as probability distribution threshold and **iterations** as the number of trails for the speaker-listener communication. For the brevity of the report, only few performance measures have been listed and compared here. Please refer **"performance and goodness metrics"** folder for the more detail performance metrics of the graphs listed above.

Goodness Measures : The goodness metrics on the communities detected is listed with respect to the experiments conducted for the amazon, youtube and dblp medium and small graphs.

Graph	#Ground Truth Communities	#Communities Detected	Average Conductance	Average Cohesive	Average Separability
Amazon Small	216	276	0.07523535	0.43538811	21.11290933
Amazon Medium	415	441	0.07408427	0.38864285	20.20078991
Youtube Small	866	870	0.0337953	0.73288426	2.21741564
Youtube Medium	1622	1862	0.10214836	0.70447436	2.4968389
Dblp Small	311	374	0.06137991	0.698662	16.10289967
Dblp Medium	624	697	0.05210291	0.7018845	16.17409585

In the above table, the graphs goodness metrics have been listed based on the evaluation against the ground truth communities with the communities detected using SLPA algorithm. The value of threshold r, is experimentally identified in order to detect communities close to the number of ground truth communities. For the brevity of the report, only few performance measures (conductance, separability, cohesiveness) have been listed and compared here. Please refer **"performance and goodness metrics"** folder for the more detail goodness metrics of the graphs listed above. Based on the above scoring functions, we can conclude that the SLPA detects the communities having **high cohesiveness, high separability and low conductance** which are the characteristics of well formed **crisp communities**.

Algorithm Evaluation: The goodness of the algorithm can be further discussed with the following points:

a) **Time Complexity**: The algorithm is gives linear time complexity i.e. it scales with the number of nodes in the graph. It takes O(Tn) time complexity, where T is the number of iterations and n is the number of vertices in the graph. Here, T is a small constant wrt n.

b) **Space Complexity:** O(Tn) , where T is the number of iterations and n is the number of vertices in the graph. T will be the size of the memory for each node.

c) **Accuracy, sensitivity, specificity :** Please refer above tables or "performance and goodness metrics" folder for the graphwise metrics.

d) **Randomized vs deterministic :** SLPA is not deterministic. When the tie happens in the popular node selection as per the listening rule or speaking rule, random node is selected.

e) **Parallel processing:** SLPA can provides asynchronous update[1] for the node memory update. This feature allow the algorithm to perform parallel updates on node with maximum 1 iteration loss.

f) **Parameter free:** slpa is not parameter free. It has a threshold r which is a minimum probability distribution of a label. As we decrease r, we get more communities. Hence community size is highly dependent on the parameter r. For the stopping criterion, number of iterations is also a required parameter. As per the results observed in the scientific publication, the default value of iterations is set to 20.

References :

[1] Slpa: Uncovering Overlapping Communities In Social Networks Via A, and Speaker-Listener Interaction Dynamic Process. "Jierui Xie and Boleslaw K. Szymanski." (n.d.): n. pag. Web.
[2] Ana, Ig Data. "LabelRank: A Stabilized Label Propagation Algorithm for Community Detection in Networks." Web.
[3] Zhao, Yunpeng. "Community Extraction for Social Networks." Proceedings of the National Academy of Sciences of the United States of America 108.18 (2011): 7321-326. Web.
[4] Kolaczyk, Eric D., and Gábor Csárdi. Statistical Analysis of Network Data with R. NewYork, NY: Springer New York, 2014. Web.