CONCISE FUZZY REPRESENTAION OF BIG GRAPHS
A Dimensionality Reduction Approach

Faisal N. Abu-Khzam, Amer Haj Ahmad and Rana H. Mouawi

LAU
الجامعة اللبنانية الأميركية
Lebanese American University

INTRODUCTION

The constant growth of social networks and the corresponding immense size of graph representation impose some difficulties in maintaining and manipulating such data. Thus, Our objective in this work is to transform a typical graph representation into a more compact form while still maintaining a high accuracy in answering adjacency queries for any two nodes in the graph. Our approach is based on mapping a graph representation to a k-dimensional space and answer queries of neighboring nodes by measuring Euclidean distances. The accuracy of our answers would decrease but would be compensated for by fuzzy logic, which gives an idea about the likelihood of error. This method allows for reasonable representation in memory while maintaining a fair amount of useful information.

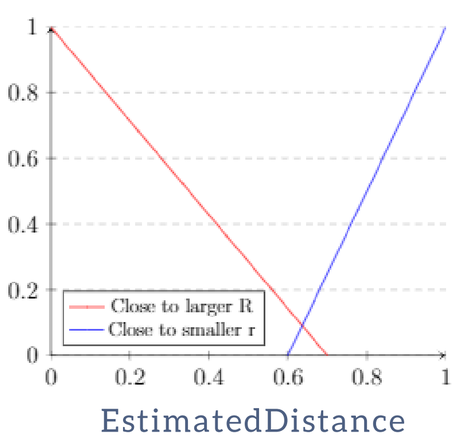
A GRAPH MAPPING APPROACH

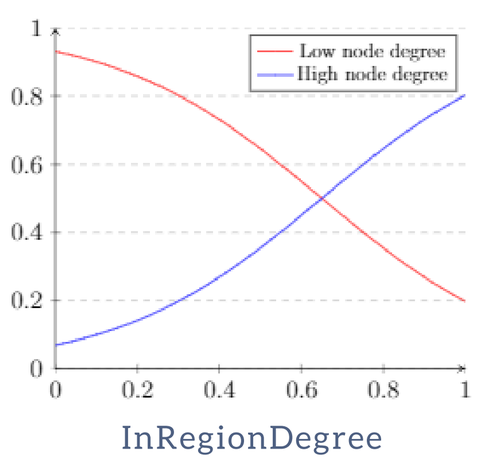
Our proposed method is to represent the nodes as points/vectors in a k-dimensional space, where k is a fixed constant, with distances between them indicating their adjacency status. To answer adjacency queries, each node is associated with two parameters, r and R, that indicate the distance (or "radius") within which its neighbors are located and the distance beyond which its non-neighbors are located, respectively. There is uncertainty regarding the nodes that lie within these two values. For the mapping part, we propose to use the linear-time FastMap algorithm (centralized and distributed versions) [1,2], which takes a distance matrix and returns a set of points in a k-dimensional space where k is user defined. Once we have this mapping, it is possible to calculate the two parameters r and R of each node.

When the Euclidean distance d between two nodes v and v′ lies within the values of r and R, we invoke a fuzzy logic system that gives us the likelihood of those two nodes being neighbors.

The system takes as input two crisp values between 0 and 1. The first value is obtained by dividing the difference of R and d by the difference of R and r. Whereas, the second value is obtained by dividing the number of neighbors between r and R by the number of overall nodes in this fuzzy region. The inputs are subjected to three membership functions that correspond to the likelihood of the queried nodes to be neighbors. The inference system relies on two (successive) rules: (1) if the first input value EstimatedDistance is "close to smaller r" AND the second input value InRegionDegree is "High node degree", then the two nodes are neighbors. If this is not the case, and (2) if the first input value is "close to larger R" AND the second input value is "Low node degree", then the two nodes are not neighbors. Note that the AND operator in the fuzzy logic inference system corresponds to taking the minimum value.

INPUT MEMBERSHIP FUNCTIONS





ADJACENCY QUERYING ALGORITHM

Algorithm 1 Adjacency Query

if $d \leq r(v)$ **or** $d \leq r(v')$ **then**
  return 1
else if $d \geq R(v)$ **or** $d \geq R(v')$ **then**
  return 0
end if
$EstimatedDistance1 = (R(v) - d)/(R(v) - r(v))$
$InRegionDegree1 = NeighborsInRegion(v)/NodesInRegion(v)$
$output1 = \text{invokeFuzzyLogicSystem}(EstimatedDistance1, InRegionDegree1)$
$EstimatedDistance2 = (R(v') - d)/(R(v') - r(v'))$
$InRegionDegree2 = NeighborsInRegion(v')/NodesInRegion(v')$
$output2 = \text{invokeFuzzyLogicSystem}(EstimatedDistance2, InRegionDegree2)$
return $\text{minimum}(output1, output2)$

EXPERIMENTS

Experiments were conducted on a number of Facebook graphs. The results reported in the below table show that the fuzzy accuracy along with the overall accuracy decrease when the graph gets denser and vice versa.

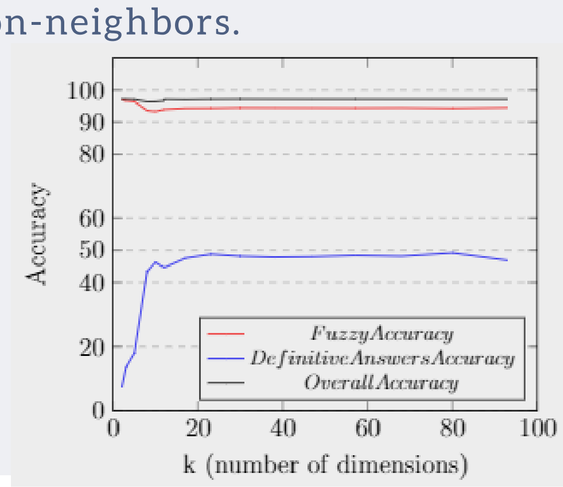
Nodes	Edges	Fuzzy accuracy	Overall accuracy
1518	32988	96.91%	97.13%
3482	155043	96.87%	97.41%
1000	200000	71%	82%
769	16656	93.29%	94.4%
2252	84387	95.7%	96.66%
4047	204850	96.33%	97.44%
6386	217662	98.08%	98.89%
1446	59589	91.57%	94.27%

GRAPH COMPRESSION

The compression ratio for our technique could reach up to almost 99.9% for large *n* being a stable ratio that could be calculated from the simple formula (n−k)/n, where *k* is the number of dimensions. However, there is a trade off between the accuracy and the compression ratio when compressing dense graphs.

RESULTS

In the figure below, the graph with the blue line shows the percentage of definite answers maintained after compression as a function of k. While the definite answers' results suggest a tremendous loss of information, fuzzy answers can almost always make up for more certain ones. A fuzzy answer is considered "sound" if it returns a value above 0.5 for a pair of nodes that happen to be neighbors, and a value below 0.5 for a pair that are non-neighbors.



REFERENCES

[1] F. N. Abu-Khzam, N. F. Samatova, G. Ostrouchov, M. A. Langston, and A. Geist. Distributed dimension reduction algorithms for widely dispersed data. In International Conference on Parallel and Distributed Computing Systems, PDCS, pages 167–174.IASTED/ACTA Press, 2002.

[2] C. Faloutsos and K.-I. Lin. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In Proceedings of the 1995 ACM SIG-MOD International Conference on Management of Data, pages 163–174, 22–25 1995.