Influential Node Prediction

# Problem Statement

Mining the most influential nodes in a network is NP-hard, while existing heuristics to extract the most influential nodes in a network do not always perform better than other heuristics in comparison. These also suffer from scalability issues.

# Study Design

## Networks

A combination of 3 types of networks were used during the study:

1. Synthetic networks:
   1. Small world
   2. Scale-free
   3. Scale-free-small world
2. Cited networks:
   1. Karate club
   2. World trade
   3. Nematode neural network
   4. Political blog
   5. Protein Network
3. Generated networks:
   1. Citation network
   2. Twitter Network

While creating synthetic networks, 6 random samples were taken against each size to avoid results by chance.

The citation network consists of a network of citations of all papers on the topic of influence mining.

Twitter network is the network of users connected to second degree from target user ID: seekme\_94

## Data Generation

Against the corpus of synthetic graphs, following traits were calculated:

1. Centrality measures:
   1. Degree
   2. Betweenness
   3. Closeness
   4. Eigenvector
   5. Eccentricity
2. Heuristics:
   1. Coreness
   2. Pagerank
   3. Collective Influence score
3. Graph traits:
   1. Graph size
   2. Number of edges
   3. Average degree
   4. Highest degree
   5. Average path length
   6. Clustering coefficient
   7. Diameter
   8. Density
   9. Assortativity
   10. Average distance
   11. Triads
   12. Girth

Some of the traits were normalized as required to standardize all data. The data was split in 50-50 for training and evaluation.

We also collected four real-world scale-free networks with various sizes and power law coefficients to test the optimal model:

1. Author NetScience
2. ITA2000
3. AS-CAIDA
4. JDK Dependencies
5. Wordnet

## Machine Learning Models

In order to pick the right model, a baseline approach of selecting the one with highest accuracy on evaluation set. The results from the models experimented upon were found to be averaging:

1. Logistic Regression: 96.413%
2. C5.0 Decision Trees: 96.388
3. Recursive Partitioning: 96.513%
4. Random Forests: 96.513%
5. Support Vector Machines: 96.504%
6. eXtreme Gradient Boost: 97.848%

The graph illustrates performance over different sizes of graphs.

Being the top performer, we picked XGBoost as our target model.

## XGBoost Resilience

The resilience from the individual traits and the XGBoost model revealed that the trained model outperforms the other heuristics in finding the most vital nodes in the network as the following table explains.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **network** | **degree** | **betweenness** | **closeness** | **eigenvalue** | **pagerank** | **eccentricity** | **coreness** | **ci** | **XGBoost** |
| author | 326 | 325 | 328 | 332 | 326 | 349 | 332 | 268 | 295 |
| ita2000 | 3036 | 3019 | 3051 | 3031 | 3019 | 3079 | 3037 | 3048 | 3013 |
| ascaida | 23834 | 23994 | 24157 | 24224 | 23880 | 22756 | 23834 | 23887 | 18580 |
| jdklibs | 5586 | 5619 | 5561 | 6101 | 5594 | 6095 | 5624 | 5592 | 4948 |