Information Retrieval and Extraction

Major Project

Community Detection from Research Articles

Hitesh Sharma (201301065)
Kanika Kanwal (201505526)
Tummalapalli Madhuri (201325191)

Problem Statement

Community detection is an important aspect in discovering the complex structure of social networks, and is of great importance in sociology, biology and computer science, disciplines where systems are often represented as graphs. A social network graph is a representation of the real world social network with the nodes representing the participating entities, or in this case, research papers or their authors, and the relations between these entities are represented by edges between them. The community detection involves grouping of similar users into clusters, where users in a group are strongly bonded with each other than the other members in the network.

In this project, “Community Detection from Research Articles”, the task then becomes detecting research papers which belong to a common field of research.

Network Communities

A network is said to have community structure if the nodes of the network can be easily grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally. In the particular case of *non-overlapping* community finding, this implies that the network divides naturally into groups of nodes with dense connections internally and sparser connections between groups. But *overlapping* communities are also allowed. The more general definition is based on the principle that pairs of nodes are more likely to be connected if they are both members of the same community(ies), and less likely to be connected if they do not share communities.

A network community has the following properties -

(1) *Mutuality of ties* - Everyone in the group has ties (edges) to one another.

(2) *Compactness* - Closeness or reachability of group members in small number of steps, not necessarily adjacency.
(3) *Density of edges* - High frequency of ties within the group.
(4) *Separation* - Higher frequency of ties among group members compared to non-members.

*Graph cliques as a community***.** A clique is a complete subgraph, i.e, a set of vertices where each pair of vertices are connected. However, cliques are a very strict definition for communities, and also computationally hard to compute.

Community detection is the task of assigning each node/vertex to a community.

Network Graph construction

For this project, we constructed a network graph from the available data using two weak metrics, and two strong ones. The weaker metrics includes using the paper titles and year of publishing as a measure of similarity. Whereas, the stronger metrics includes paper citation and author citation.

- Network based on similarity of **paper titles**
    - In this case, the network graph for the research papers was constructed using the paper titles. This was done by first constructing a similarity matrix between each pair of papers. Then, a threshold was used in order to decide whether an edge must be exist between a pair of nodes or not. In case of the weighted graph, the measure of similarity between the nodes was used as the weight.

- Network based on **year** when the paper was published
    - The network graph, for this case, was constructed in a similar manner as in the case of titles. An edge is present between two papers if they are published in the same year. This leads to the formation of separate cliques in the network graph, and so the communities are obtained naturally.

- **Paper citations** network
    - Paper citation network is basically a network that is constructed on the basis of the citations made in a paper to another paper. That is, if a paper refers to another paper, we have an edge linking the two papers. When building a weighted graph, the weight of the edge is equivalent to the number of times a citation repeats itself.

- **Author citation** network
    - Author citation network is a network that is based upon the various authors who cite each other's work. Or, we can say, that an edge will exist between two nodes, in this case, authors, if the two of them have ever cited each other's paper. If they have worked together multiple times, means more the number of citations then the weight on this edge is increased proportionally.

Finding the Communities

Finding network communities can be seen as a -

- Graph clustering problem
- Graph partitioning problem

Community Detection as a Graph clustering problem

This technique was applied on the weaker metrics, namely paper title and year of publication.

Naive definition for a community -

Network communities are group of vertices similar to each other.

Similarity based vertex clustering can be briefly described as follows -

- Define similarity measure between vertices based on network structure and calculate similarity between all pairs of vertices in the graph, in this case, we used Cosine similarity and Jaccard similarity measures for the purpose.

    - **Cosine Similarity :** For each of the weaker metrics, we defined a similarity measure between each pair of nodes, in order to construct the network. Thus we obtain a vector representation for each node (paper) where the ith value represents its similarity with the ith node.

      Cosine Similarity can then be calculated as:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

    Here, ***A*** and ***B*** are the vectors whose similarity needs to be found.

    - **Jaccard Similarity :** To find the Jaccard similarity, we first find the neighbour set for each node in the graph and then find the similarity as:

      $$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

      Here, ***A*** and ***B***, are the neighbour sets of the two nodes. A neighbour set for any node is defined as the set of all nodes linked to it in the network/graph.

- Grouping the similar vertices together using some clustering algorithm. For our purpose, we used the K-Means clustering algorithm.
    - **K-Means Clustering Algorithm -** The K-Means clustering algorithm is an Expectation-Maximization algorithm. It first chooses K nodes or centroids at random, and then iterates repeatedly through the expectation step and the maximisation step.

      In the **expectation step**, it assigns each of the nodes to the centroid that is the most similar to it.

      In the **maximization step**, a new centroid is found for each of the clusters so formed. While using cosine similarity, it is found as the average of the nodes in the cluster. And for jaccard similarity, the node with the maximum number of edges in that cluster is made the new centroid.

Problems

- Metrics such as paper titles, and year of publication are based on similarity measures, and are not an intuitive representation of the research articles network. Better metrics such as author citations and paper citations can be used to construct the network.
- When using K-Means, the number of possible communities that exist in the network must be known or approximated before hand. This number is provided as input to the K-Means algorithm. This approximation can be incorrect and hence result in insufficient network communities.
- K-Means might converge to a local maxima, and therefore needs to be restarted to give better results.

Community Detection as a Graph partitioning problem

This technique was applied on the stronger metrics of paper citations and author citations. Graph partitioning algorithms find communities based on the structural similarity of nodes in the graph.

An alternative definition for network communities -

> *Network communities are groups of vertices such that the vertices inside the group are connected with many more edges than between the groups.*

Algorithms used -

- Newman-Girvan algorithm (based on edge-betweenness)
- Louvain algorithm (based on maximising modularity)

Newman-Girvan algorithm (Edge-Betweenness)

Introduction

Newman-Girvan algorithm focuses on the edges that connect communities. It is based on the simple idea that the shortest path between nodes from different communities will always include the edges that connect the different communities.
Betweenness value for any edge is calculated as:

$$C_B(e) = \sum_{s \neq t} \frac{\sigma_{st}(e)}{\sigma_{st}}$$

Algorithm

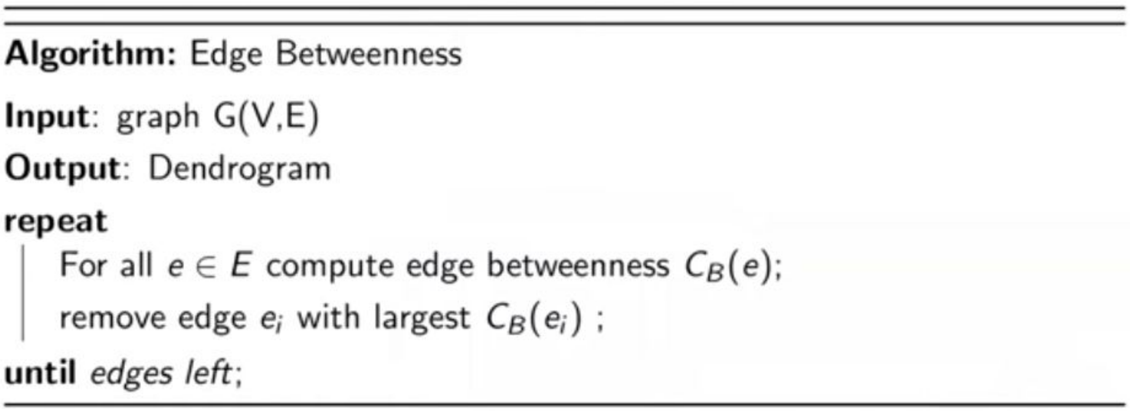
1) Find the edge of highest betweenness - or multiple edges of highest betweenness. Remove these edges from the graph. This may cause the graph to separate into multiple components. If so, this is the first level of regions in the partitioning of the graph.
2) Now recalculate all betweenness, and again remove the edge or the edges of highest betweenness. This may break some of the existing components into smaller components.
3) Repeat the above steps as long as edges remain in the graph.

As the algorithm proceeds, edges are removed from the graph, and so the graph splits into different connected components. These connected components can be thought of as different communities. In each iteration of the algorithm, we find the modularity value for the connected components (communities). The set of connected components with the highest modularity value is the final desired output.

The modularity score is used as a measure for community “quality”. Higher the modularity score, the better the communities. The modularity score ranges between [-0.5, 1), and is calculated as:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j), \quad \delta(c_i, c_j)\text{- kronecker delta}$$

Pseudo Code



```
Algorithm: Edge Betweenness
Input: graph G(V,E)
Output: Dendrogram
repeat
    For all $e \in E$ compute edge betweenness $C_B(e)$;
    remove edge $e_i$ with largest $C_B(e_i)$ ;
until edges left;
```

Drawbacks

- Finding the edge betweenness is a computationally expensive step, and hence the algorithm can not be used for large real world networks and is not scalable.

Louvain Method (Maximising modularity)

Introduction

The Louvain algorithm partitions the graph by optimising the **graph modularity measure.** Modularity is a scale value between -1 and 1 that measures the density of edges inside communities to edges outside communities. For weighted graphs, this measure can be calculated as:

$$Q = \frac{1}{2m}\Sigma_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j)$$

A_{ij} represents the edge weight between nodes i and j. k_i and k_j are the sum of the weights of the edges attached to nodes i and j respectively. m is half the sum of all edge weights in the graph. c_i and c_j are the communities of the nodes, and δ is a simple delta function.

Optimising this value results, theoretically, in the best possible partition. But, doing so is computationally expensive. Therefore, the Louvain method first optimises modularity locally, on each node, forming local communities. These local communities are then treated as a single node and the first step is repeated again.

Algorithm

The algorithm iterates, repeatedly through the following two steps, until the change in modularity is close to nill.

- Initially, every node is assigned to its own community. Then the change in modularity is calculated when the node is removed from its own community, and added to its neighbours community. This is done for every neighbour of the given node. Finally, the node is assigned to the community whose change in modularity is the maximum. In case the maximum change is negative, the node is not added to any community. The change in modularity is calculated using :

$$\Delta Q = \left[\frac{\Sigma_{in} + k_{i,in}}{2m} - \left(\frac{\Sigma_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

Where Σ_{in} is sum of all the weights of the links inside the community i is moving into, Σ_{tot} is the sum of all the weights of the links to nodes in the community, k_i is the degree of i, $k_{i,in}$ is the sum of the weights of the *links between i and other nodes in the community, and m is half the sum of the weights of all links in the network.*

- In the second step, all the nodes in a single community are grouped together, and a new network is formed where the nodes are the communities from the previous phase.

The algorithm terminates when there is no more change in the modularity score.

Advantages
The biggest advantage of this algorithm is that it finds the communities in *O(nlogn)* time, whereas most of the other algorithms take at least $O(n^3)$ time. Therefore, even for large networks with about 20,000 nodes, this algorithm finds the communities in about 6 secs.

Disadvantages
This algorithm has the disadvantage of needing to store all the nodes of the community in the main memory.

Experiments Conducted and Analysis

The following table lists the experiments conducted:

Experiment Number	**Metric Used**	**Algorithm**
1	Paper Titles	KMeans with Cosine Similarity
2	Paper Titles	KMeans with Jaccard Similarity
3	Year of Publication	KMeans with Cosine Similarity
4	Year of Publication	KMeans with Jaccard Similarity
5	Paper Titles (weighted and unweighted)	Louvain Algorithm

6	Paper Citations (weighted and unweighted)	Newman-Girvan Algorithm
7	Paper Citations (weighted and unweighted)	Louvain Algorithm
8	Author Citations (weighted and unweighted)	Newman-Girvan Algorithm
9	Author Citations (weighted and unweighted)	Louvain Algorithm

Analysis

- The year metric was the least effective of all, since, with this metric every paper that was published in a particular year automatically falls into the same community. But, it is easy to observe that this is not the case (two papers published in the same year need not be related).

- The title metric was seen to perform better than the year metric, but still it was not very efficient. It was visible from the output, that papers with titles containing common terms in them were often grouped together. But the assumption that papers with similar titles will be similar in their content was found to be wrong.

- Both K-Means and Louvain were used on the paper titles metric. The greatest benefit of Louvain over K-Means was that with K-Means, we had to specify the number of communities apriori. But this value is not known.

- When using Paper citations as a metric, the results are seen to improve a lot, and is can be seen that works belonging to a community are similar in their content. Thus, the clusters formed using this technique seem to be much similar than before. Even the Author citations proved to be a good metric.

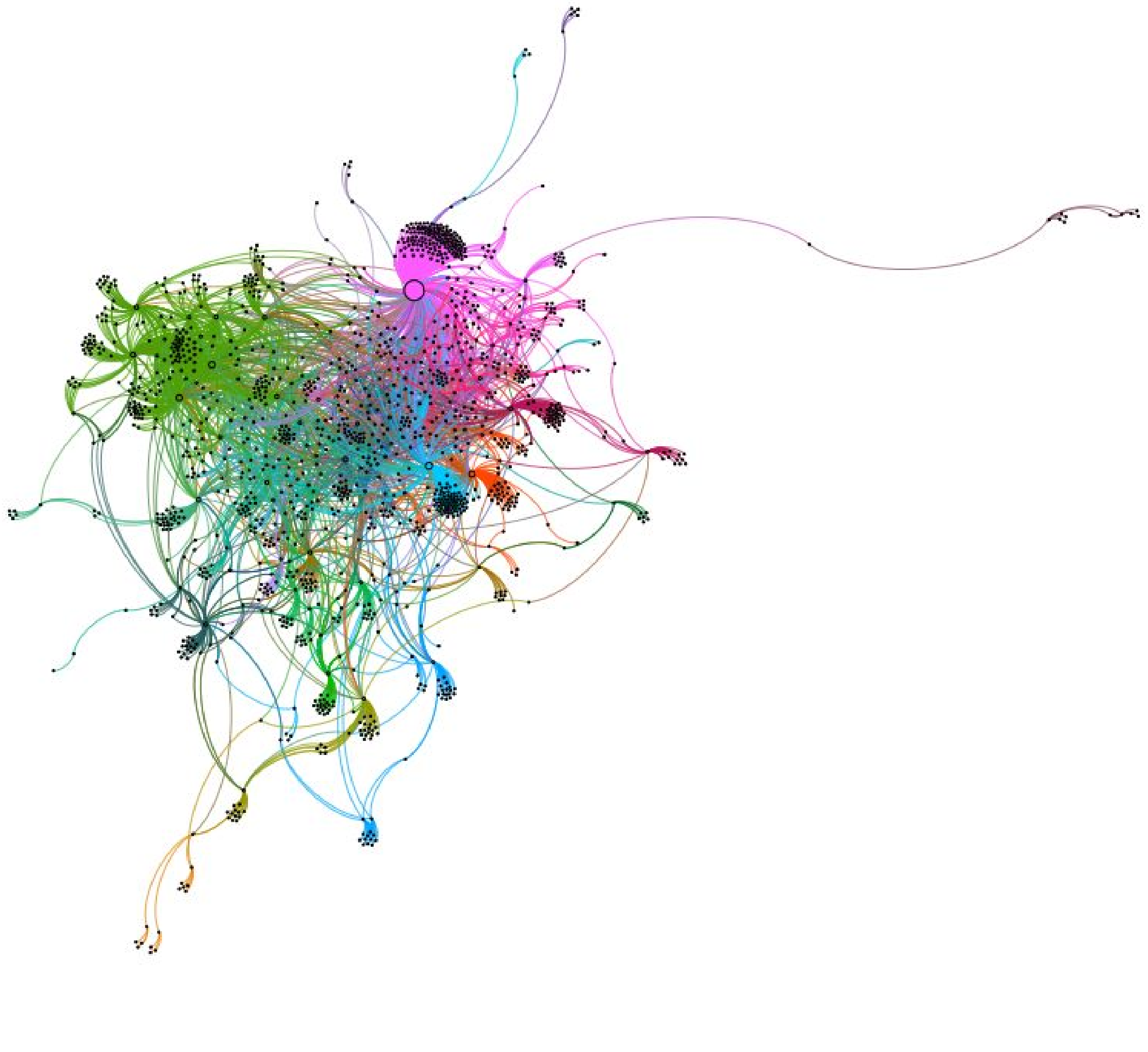
- Both Newman-Girvan and Louvain have the benefit of not requiring the number of communities as an input. But, the difference between these two was that Newman-Girvan was seen to be much slower than Louvain. (Louvain took approx 6 secs on a network of about 20,000 nodes, Newman-Girvan took hours on 3000 nodes itself).

Plots

Each community is assigned a unique colour shade.

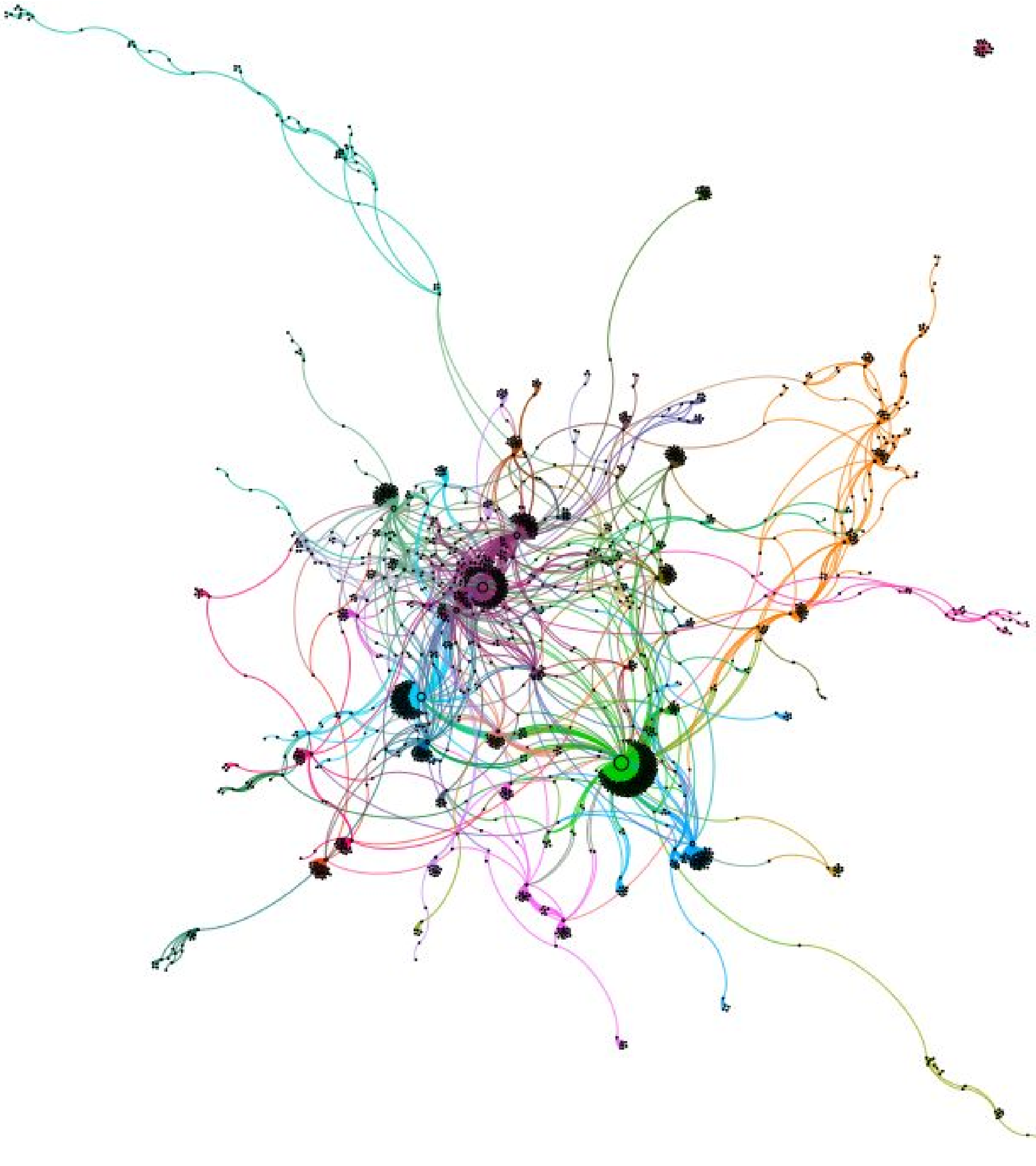
Newman-Girvan

- Unweighted author citation network (2500 nodes)
    - Communities: 34



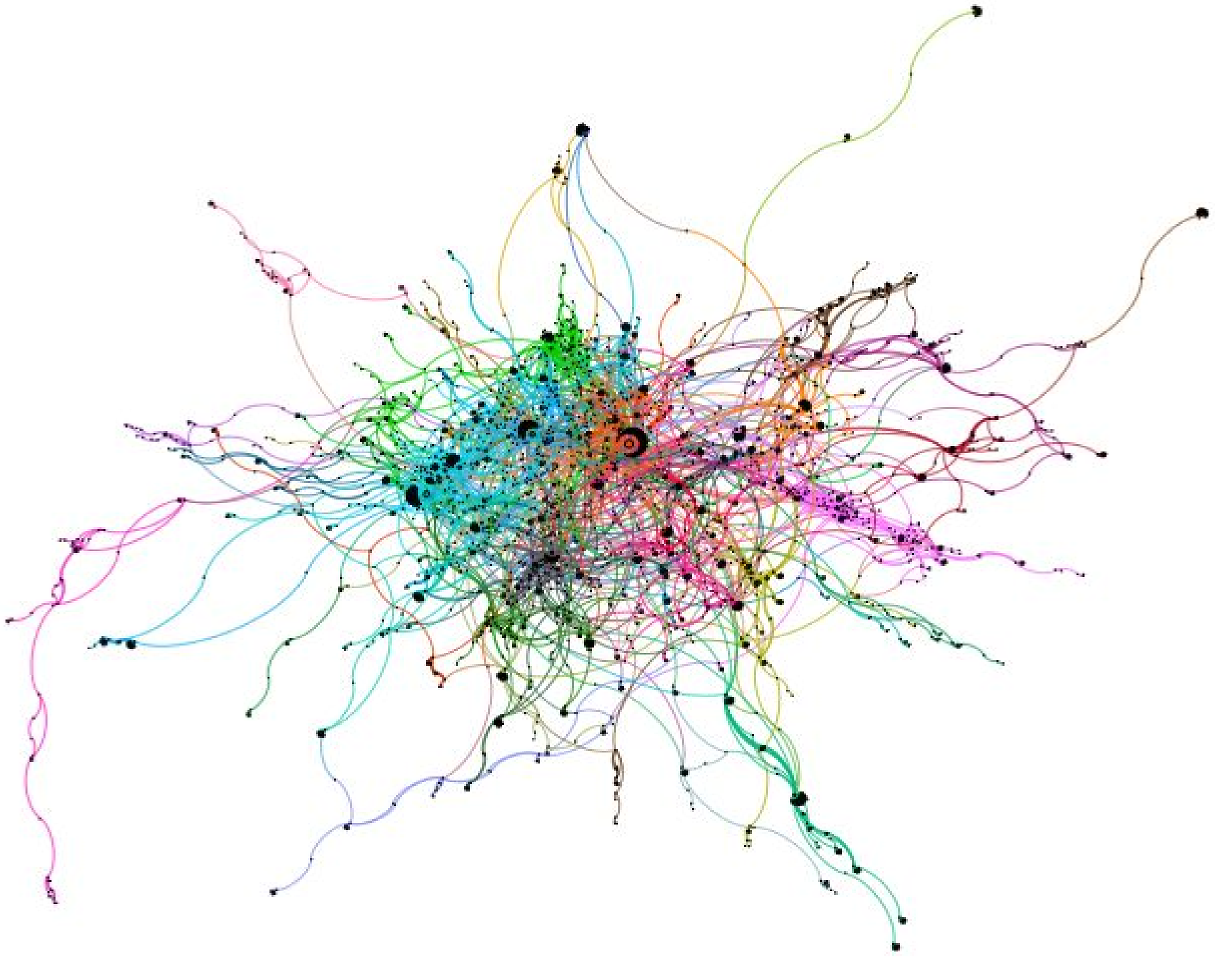
Newman-Girvan

- Unweighted paper citation network (2500 nodes)
  - Communities: 101



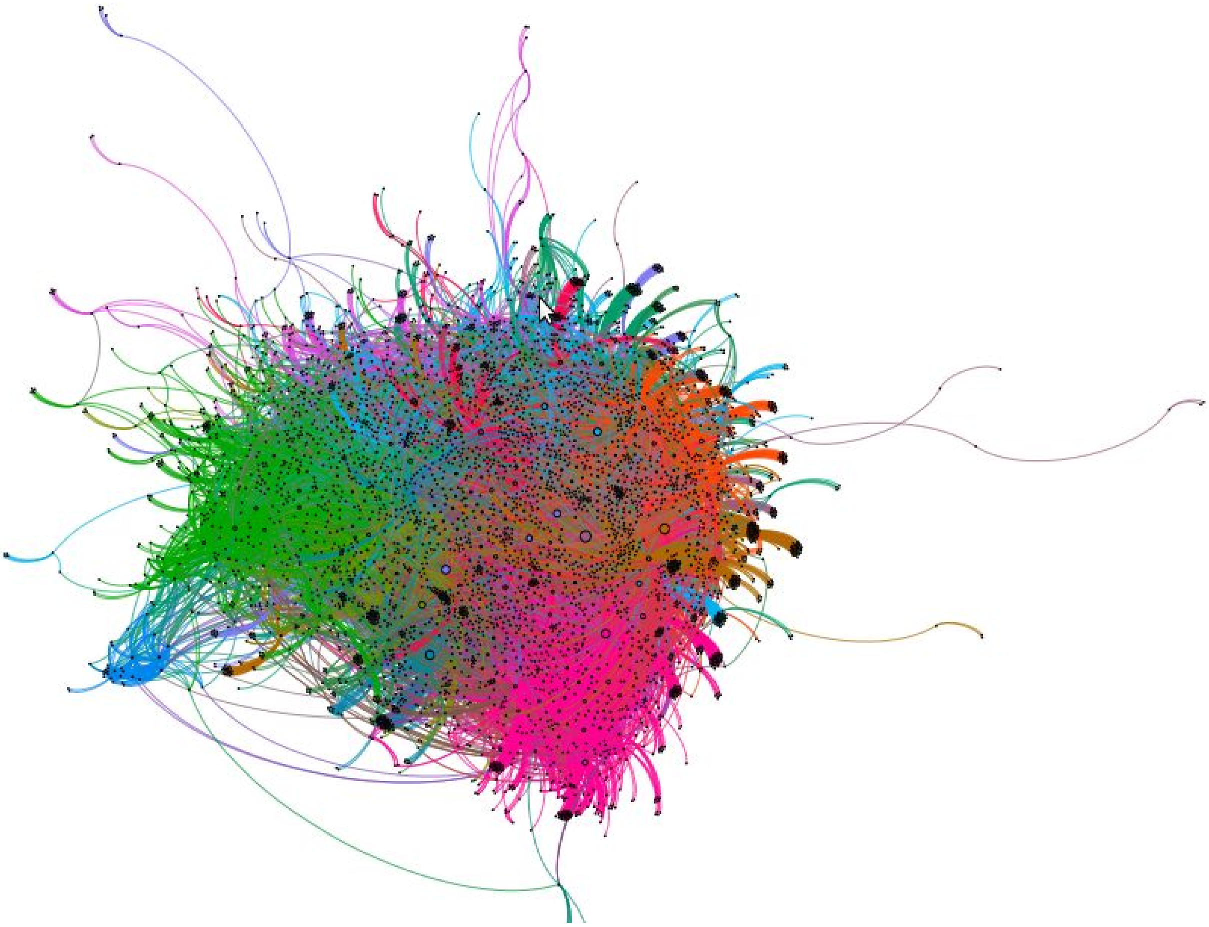
Louvain

- Weighted paper citation network (5000 nodes)
    - Communities: 180



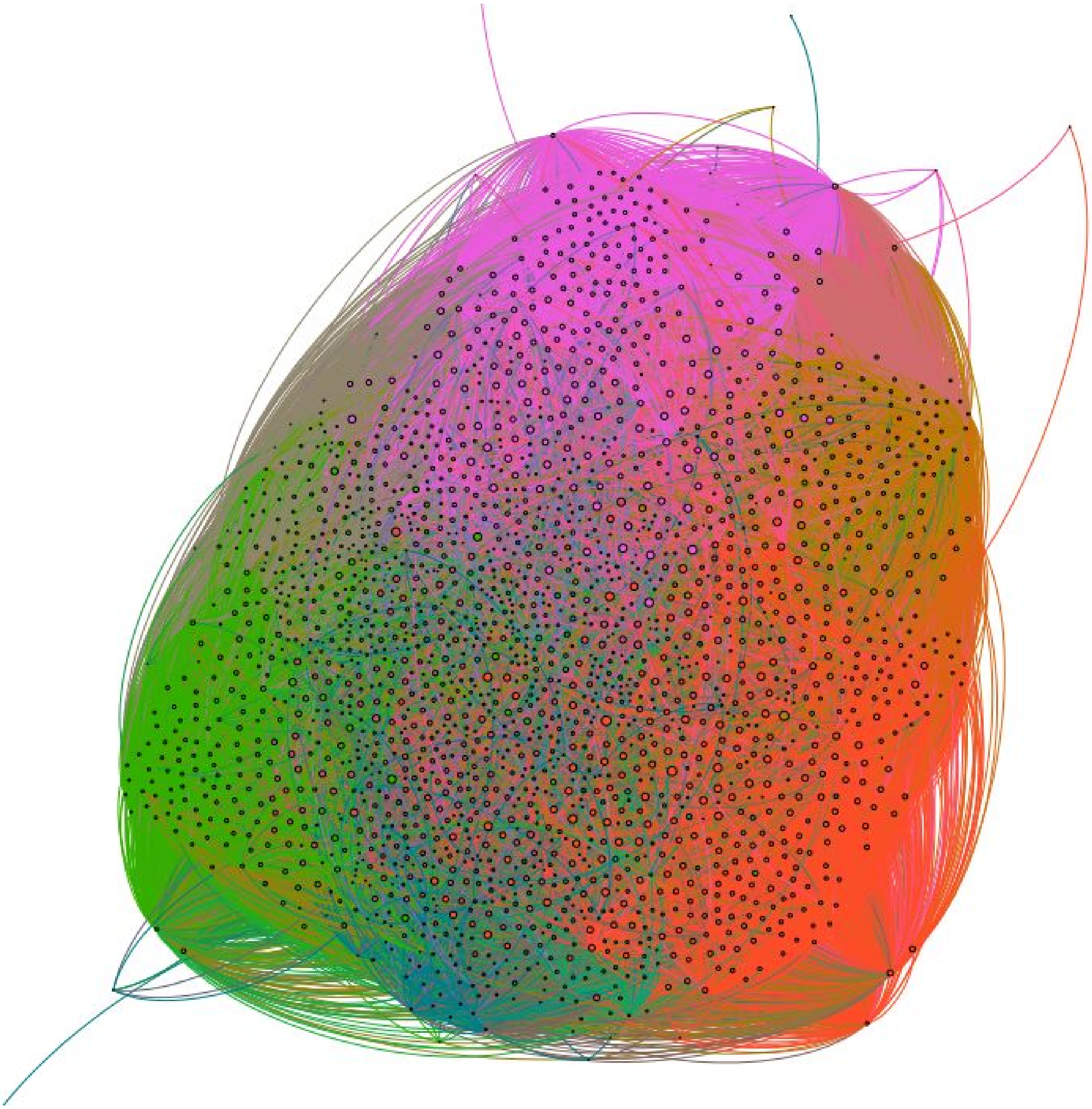
Louvain

- Weighted author citation network (5000 nodes)
    - Communities: 27



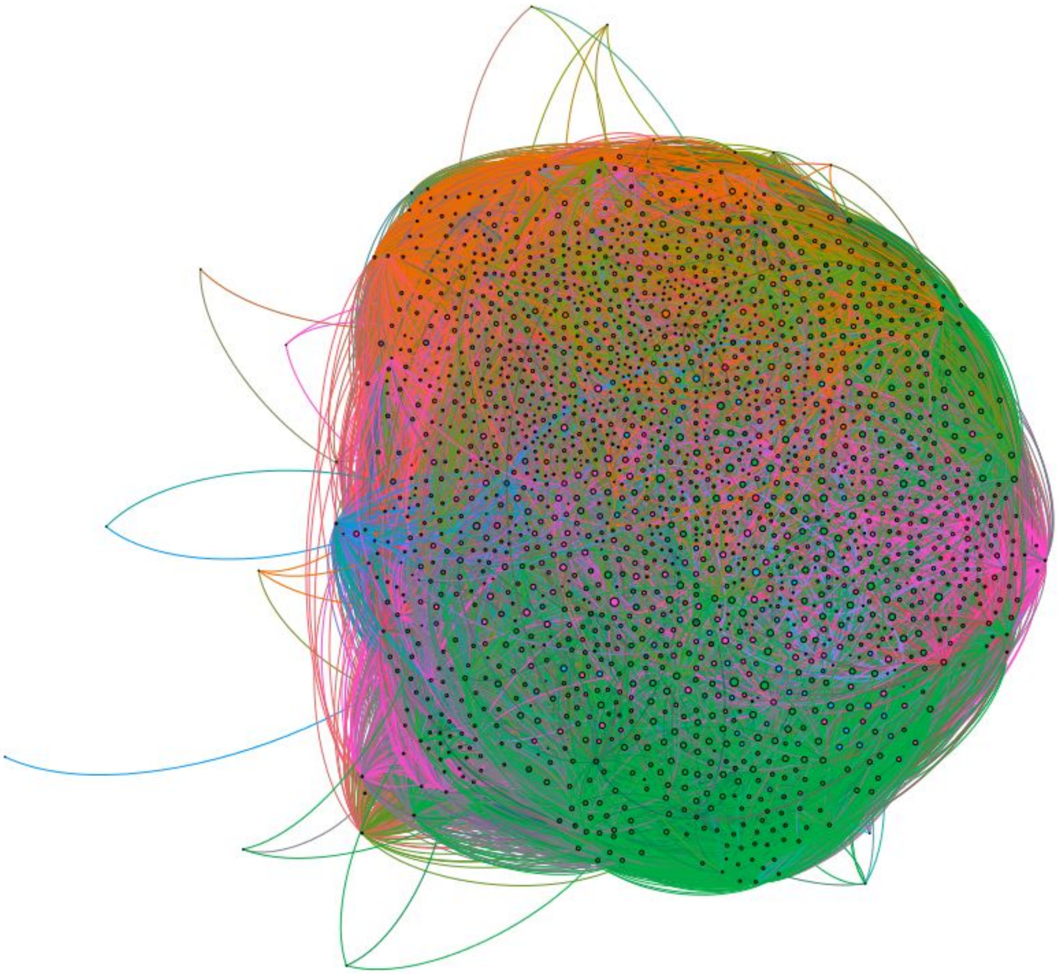
Louvain

- Unweighted graph based on titles (2000 nodes)
  - Communities: 10



K-Means

- Unweighted graph based on titles (2000 nodes)
    - Communities: 20



Dataset

- AAN Dataset
  - http://clair.eecs.umich.edu/aan/index.php

References

- *Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte and Etienne Lefebvre;* Fast unfolding of communities in large networks. *(2008)*
- M. E. J. Newman; Detecting community structure in networks.
  - http://www-personal.umich.edu/~mejn/papers/epjb.pdf
- Gephi tool
  - https://gephi.org/
- *Charu C. Aggarwal, Yan Xie and Philip S. Yu;* Towards Community Detection in Locally Heterogeneous Networks
- *https://en.wikipedia.org/wiki/Louvain_Modularity*
- *https://en.wikipedia.org/wiki/K-means_clustering*
- *https://www.youtube.com/watch?v=lU1QEUH0nNc*