**Modularity and community structure in networks**  
**Requirements to run the program**

**networkx**

**matplotlib**

**scipy**

**numpy**

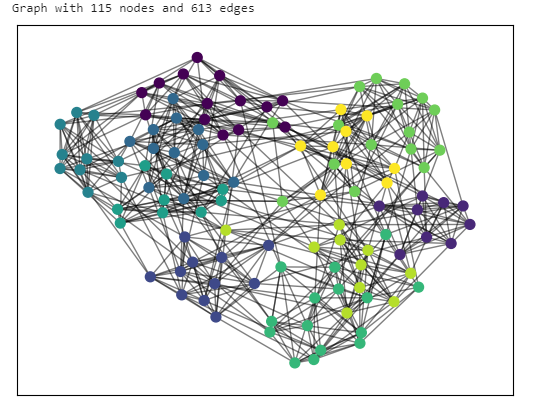
**sklearn**

**python-louvain  
  
  
Dataset used**

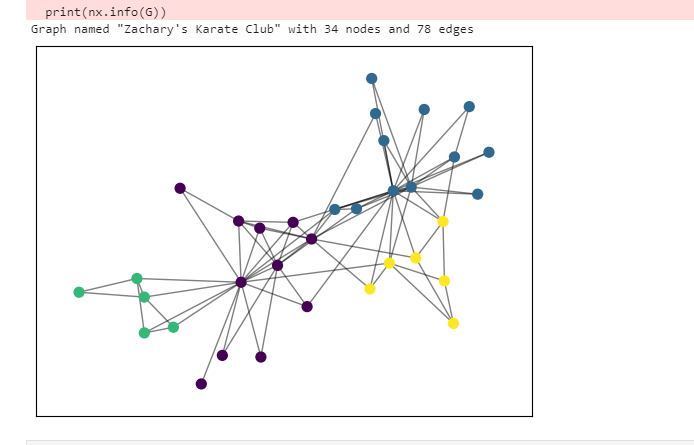
* **The American College football network dataset**
* **The zachary'karate dataset**
* **The Bottlenose Dolphins network dataset**
* **The Books about US politics networks dataset**
* **The US power grid network**
* **The Email, the university network as a sparse adjacency matrix**

**Note: It is important to download the dataset, create a new folder called inputs and extract all the datasets in that new folder for easy usage.**

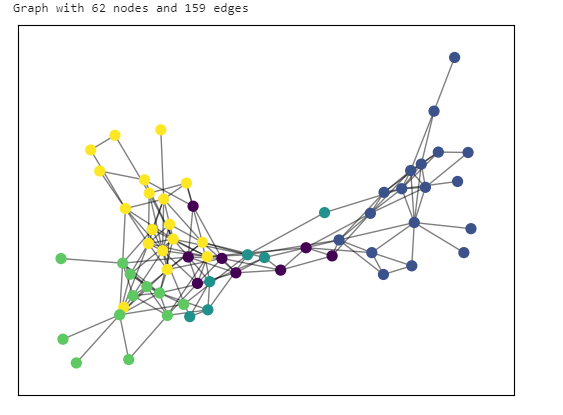
The first section focuses on community detection using the Louvain algorithm and visualizing the results. It begins by importing the required libraries such as **networkx**, **matplotlib**, and **community**. Then, it loads various network datasets, such as the American College football network, Zachary's karate club dataset, Bottlenose Dolphins network, Books about US politics network, and US power grid network. Only one network dataset is loaded at a time by uncommenting the corresponding line.

After loading the network dataset, the code uses the Louvain algorithm (**community\_louvain.best\_partition()**) to compute the communities within the network. The detected communities are stored in the **partition** variable. Next, the code visualizes the network with nodes colored according to their community assignments. The spring layout algorithm is used to position the nodes, and the **nx.draw\_networkx\_nodes()** and **nx.draw\_networkx\_edges()** functions are used to draw the network.  
The figure for The American College football network dataset  


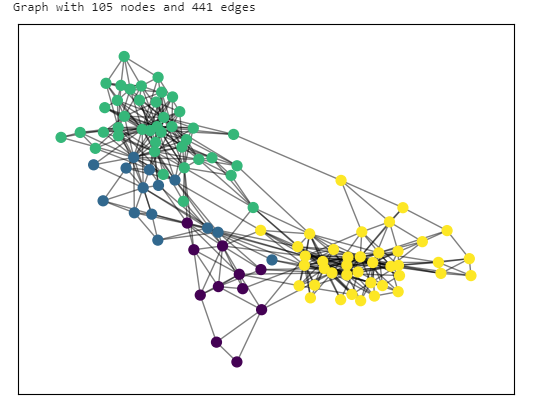
The figure for The zachary'karate dataset



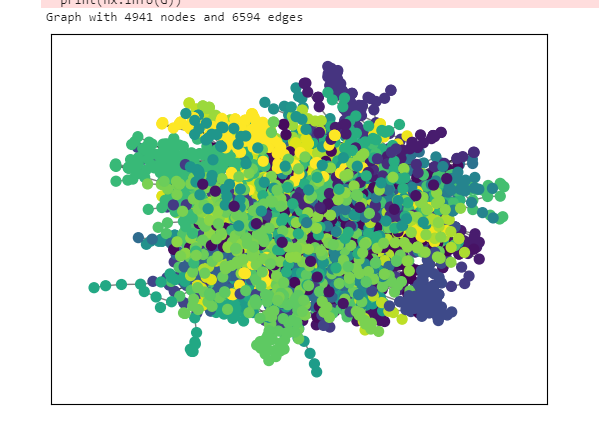
The figure for The Bottlenose Dolphins network dataset



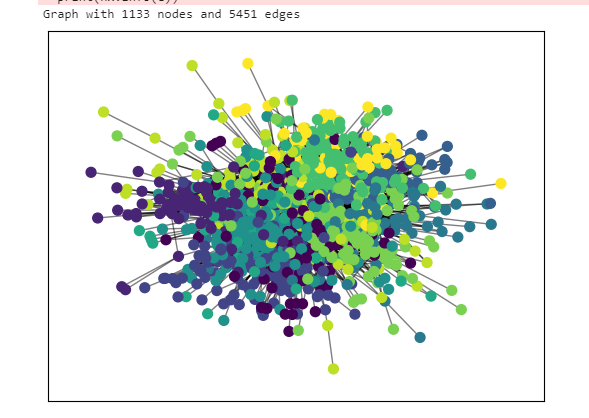
The figure for The Books about US politics networks dataset



The figure for The US power grid network



The figure for The Email, the university network



Section 2: Community Detection using Louvain Algorithm and Evaluation Metrics

The second section also focuses on community detection using the Louvain algorithm but includes the evaluation of the detected communities. It starts by importing the required libraries, including **networkx**, **community**, and **numpy**. Similar to the previous section, different network datasets can be loaded by uncommenting the corresponding lines.

Once the network dataset is loaded, the code applies the Louvain algorithm (**community.best\_partition()**) to detect communities within the network. The detected communities are stored in the **partition** variable. The code then prints the number of communities detected and lists the nodes belonging to each community.  
 **Number of communities recorded in each dataset used**  
The American College football network dataset

Graph with 115 nodes and 613 edges

Number of communities: 10

The zachary'karate dataset

Graph named "Zachary's Karate Club" with 34 nodes and 78 edges

Number of communities: 4

The Bottlenose Dolphins network dataset

Graph with 62 nodes and 159 edges

Number of communities: 6

The Books about US politics networks dataset

Graph with 105 nodes and 441 edges

Number of communities: 4

The US power grid network

Graph with 4941 nodes and 6594 edges

Number of communities: 39

The Email, the university network

Graph with 1133 nodes and 5451 edges

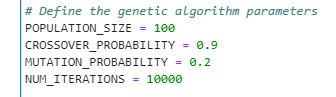
Number of communities: 11

Section 3: Genetic Algorithm for Community Detection

The third section introduces a genetic algorithm approach for community detection. It begins by importing the necessary libraries, including **community**, **networkx**, and **random**. The code loads the Zachary's karate club dataset, which can be changed to other network datasets by uncommenting the corresponding lines.

The code defines a fitness function (**fitness()**) to evaluate the quality of community structures. It then initializes a population of random community structures and evaluates their fitness. The code performs elitism by selecting the top 10% of the population based on their fitness scores and carries them over to the next generation. It creates the next generation of the population using tournament selection, crossover, and mutation operations. The process continues for a specified number of iterations, and the best modularity value found so far is printed at each iteration.

Variable used:



Note it is important to set a higher number of iteration to produse a maximum modularity value, at some point you will realize that after reaching a maximum possible modularity value, the iteration will continue but repeating the modular value which has been achieved to be the highest

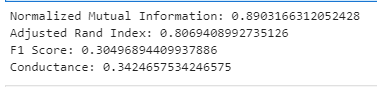
We then optimized the values of modularity to explore other possible outcome using greedy algorithm, then we further used Louvin algorithm to get the highest possible modular value.

While incorporating the three techniques, that is Greedy, Louvin Algorithm and leveraging Elitism techniques for modularity optimization, this is the final maximum possible modularity that was recorded

|  |  |
| --- | --- |
| Dataset | Modularity value |
| The American College football network | 0.6044 |
| The zachary'karate club | 0.4449 |
| The Bottlenose Dolphins network | 0.5233 |
| The Books about US politics networks | 0.5269 |
| The US power grid network | 0.9355 |
| The Email, the university network | 0.5760 |

**The following metrics was used for evaluation**

1. Normalized Mutual Information (NMI): NMI measures the agreement between the true labels of the nodes and the predicted community assignments. A value **of 0.890** indicates a relatively high level of similarity between the true and predicted communities. It suggests that the algorithm has successfully captured the underlying community structure in the network.
2. Adjusted Rand Index (ARI): ARI is another measure of agreement between the true and predicted communities, taking into account chance agreement. An ARI value **of 0.807** indicates a substantial agreement between the true and predicted communities. It suggests that the algorithm's results are significantly better than random assignment.
3. F1 Score: The F1 score measures the accuracy of the community detection algorithm in terms of precision and recall. A value **of 0.305** suggests that the algorithm has achieved moderate accuracy in detecting communities. It indicates that there is room for improvement in correctly identifying nodes belonging to their respective communities.
4. Conductance: Conductance is a measure of how well a community is separated from the rest of the network. A lower conductance value indicates a more cohesive and well-separated community structure. The conductance value **of 0.342** suggests that the detected communities have a moderate level of separation from the rest of the network. It indicates that the algorithm has achieved some level of community separation, but there is room for improvement.



In summary, the results indicate that the community detection algorithm has achieved a good level of agreement with the true communities (as indicated by NMI and ARI), but there is room for improvement in terms of precision and recall (F1 score) and community separation (conductance). Further analysis and optimization may be required to enhance the accuracy and separation of the detected communities.

**References**

Elitism Techniques

1. E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms - A comparative case study," in Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature (PPSN V), 1998. [Link](https://www.researchgate.net/publication/221489883_Multiobjective_optimization_using_evolutionary_algorithms_-_A_comparative_case_study)
2. H. Ishibuchi, T. Nakashima, and M. Nii, "Comparison of selection strategies in elitist genetic algorithms," in Proceedings of the First IEEE Conference on Evolutionary Computation, 1994. [Link](https://ieeexplore.ieee.org/document/400960)
3. D. Goldberg, "Genetic algorithms in search, optimization, and machine learning," Addison-Wesley Professional, 1989.
4. C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," ACM Computing Surveys (CSUR), 2003. [Link](https://dl.acm.org/doi/10.1145/937503.937505)
5. E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, 2001. [Link](https://link.springer.com/chapter/10.1007/3-540-45356-3_27)

Metrics

1. Normalized Mutual Information (NMI):
   * Strehl, A., & Ghosh, J. (2003). Cluster ensembles: A knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research, 3, 583-617. [Link](https://www.jmlr.org/papers/volume3/strehl02a/strehl02a.pdf)
2. Adjusted Rand Index (ARI):
   * Hubert, L., & Arabie, P. (1985). Comparing partitions. Journal of Classification, 2(1), 193-218. [Link](https://link.springer.com/article/10.1007/BF01908075)
3. F1 Score:
   * Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies, 2(1), 37-63. [Link](https://jmlr.csail.mit.edu/papers/volume12/powers11a/powers11a.pdf)
4. Conductance:
   * Leskovec, J., Lang, K. J., Dasgupta, A., & Mahoney, M. W. (2010). Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. Internet Mathematics, 6(1), 29-123. [Link](https://arxiv.org/abs/0810.1355)

Datasets

1. American College football network:
   * Girvan, M., & Newman, M. E. (2002). Community structure in social and biological networks. Proceedings of the National Academy of Sciences, 99(12), 7821-7826. [Link](https://www.pnas.org/content/99/12/7821)
2. zachary'karate dataset:
   * Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. Journal of Anthropological Research, 33(4), 452-473. [Link](https://www.journals.uchicago.edu/doi/10.1086/jar.33.4.3629752)
3. Bottlenose Dolphins network:
   * Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E., & Dawson, S. M. (2003). The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. Behavioral Ecology and Sociobiology, 54(4), 396-405. [Link](https://link.springer.com/article/10.1007/s00265-003-0651-y)
4. Books about US politics networks:
   * Krebs, V. E. (2004). Mapping networks of terrorist cells. Connections, 24(3), 43-52. [Link](https://www.jstor.org/stable/43849549)
5. US power grid network:
   * Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. Nature, 393(6684), 440-442. [Link](https://www.nature.com/articles/30918)
6. Email, the University Network:
   * Guimera, R., Mossa, S., Turtschi, A., & Amaral, L. A. (2005). The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. Proceedings of the National Academy of Sciences, 102(22), 7794-7799. [Link](https://www.pnas.org/content/102/22/7794)