JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 08 Issue 01
2024

Research Article
https://doi.org/10.56979/801/2024

Experimental Analysis of Algorithms for Community Detection

Urooba Khalid[1], M. Imran Khan Khalil[1*], Asif Nawaz[2], Izaz Ahmad Khan[3], Mian M. Aimal[4], and Sheeraz Ahmed[5]

[1]University of Engineering & Technology, Peshawar, 25000, Pakistan.
[2]Higher Colleges of Technology, Dubai, UAE.
[3]Department of Computer Science, Bacha Khan University, Charsadda, 24420, Pakistan.
[4]Virtual University, Islamabad, 24420, Pakistan.
[5]Iqra National University, Peshawar, 24420, Pakistan.
[*]Corresponding Author: Muhammad Imran Khan Khalil. Email: imrankhalil@uetpeshawar.edu.pk

Received: August 21, 2024 **Accepted:** December 01, 2024

Abstract: This is an exciting new approach because by understanding the anatomy of networks, you can get a valuable framework to define related phenomena, from social and technological systems to all sorts of other complex systems found in the real world. Community structure is an essential attribute of complex networks, which has been an active field of research for decades. Understanding the structure of networks is a crucial problem, and community detection algorithms are the most utilized strategies for that. Detecting communities is fundamental for understanding their structure, function, evolution, and dynamics. As a result, the concept of community structure has attracted significant interest in recent years. When thinking about product recommendations, it is crucial to pay attention to finding sub-networks within the co-purchasing network. Keeping this in mind, we made the decision to start research that would involve analyzing four methods for community discovery. We wanted to select some real-world data for our experiment, so we used the Amazon co-purchasing network datasets to test the algorithms we had selected. In addition, we plan to investigate future work that integrates machine learning techniques with algorithms for detecting communities. In this situation, factors such as run time, efficiency, and modularity scores become significant. As a result, our project will concentrate on assessing the run time and modularity ratings of each algorithm in particular.

Keywords: Community Detection; Algorithms; Social Networks; Recommendation Systems; Machine Learning.

1. Introduction

Online retail has grown exponentially over the last decade, with companies such as Amazon putting more and more effort into recommending products [1], [2]. To improve recommender systems, a key area to focus on is the relationships between products. Networking similarities are based solely on products bought together or chosen by users with similar profiles [3]. This product-centric network is essential for influencing consumers, as they tend to be affected by the behaviour and purchasing patterns of their peers, and are naturally attracted to the goods that others have expressed interest in, thus creating trust in certain products [4].

As a result, effective product recommendation is not only a matter of implementing the right algorithm; it requires an understanding of the co-purchasing network that goes beyond the algorithm. Recognizing these relations is essential for online retailers [3]. In this context, community detection algorithms are crucial, and we contribute exciting machine learning methods to partially automate the analytical workflow [5].

Before presenting our proposed methodology based on machine learning, we would like to test four common community detection algorithms and assess their modularity and run times [6], [7], [8]. The findings will be framed within a machine learning perspective, especially reflecting on how a model with time complexity ($n2$) can grow to ($n3$) with ML techniques. It is important for us to find a community detection algorithm that works best for our investigation [9], [10]. Therefore, we will process only the run-time performance and modularity results for the selected algorithms [11].

2. Material and Methods

2.1. Proposed Algorithms

We choose the following: The Louvain Algorithm, the Infomap Algorithm, the Fast Greedy Algorithm, and the Label Propagation Algorithm. Fast Greedy is very alike Louvain so we intentionally replace it with Label Propagation in our analysis [12], [13]. We also included two others widely used algorithms from before in our experiment, Newman's and Leiden's. We will delve into these in the next section. You are an untouchable algorithm — this is what you will find from Infomap. It utilizes the information flow and employs the "map equation" to determine the most efficient method for compressing information regarding random walks on the network [14], [15]. Random Walks: Envision a "random surfer" navigating the network, leaping from node to node across established connections [16]. The surfer becomes ensnared by closely-knit networks and allocates more time to them. For instance, Infomap constructs a codebook (analogous to a dictionary) to effectively characterise these random paths [17].

- Two-Part Codes: A two-part code is assigned to each node:
- Module Code: A code identifying the community to which a specific node belongs to.
- Node Code: Node identity in its specific community.

Minimal Description Length: Infomap reduces the average code length necessary to articulate a typical random-walk trajectory. This allows us to choose communities that produce the most condensed depiction of the walks. Infomap is a prominent and robust community detection technology that use information flow to reduce the description length of random walks within a network [18]. It assigns a dual-part code to each node, encapsulating both its community affiliation and individual identity. Infomap is applicable to various networks and generates hierarchical communities. However, the accuracy of the method may diminish for larger networks over 1,000 nodes [19].

2.2. Louvain

The Louvain method is a widely utilized and effective approach for identifying communities inside networks. It is predicated on optimizing modularity, concentrating on identifying network partitions that enhance the modularity score [20]. Modularity: is a measurement used to ascertain how successful a division of a network into communities is. It analyses how many more or fewer connections there actually are in the communities than you would expect if connections were assigned randomly. The higher the score, the better the structure of the community [21]. The Louvain method employs a greedy optimization strategy. Initially, each node resides in an individual community, and subsequently, nodes are iteratively transferred across communities to enhance the modularity score [22]. The Louvain approach can identify hierarchical communities—sub-communities within larger communities. This can be recursively applied to identify communities to extract more sub-communities. Modularity optimization is a frequently used community detection method, and the Louvain method is one of the most widely-used [3] algorithms based on modularity optimization [23]. It starts with all nodes placed in a single community and greedily exchanges the respective nodes within the communities, resulting in maximized modularity. It can find the hierarchical structure of communities and is efficient in terms of time complexity [24], [25].

2.3. Label Propagation

This algorithm is a simple and very effective network community detection method. The underlying idea here is that since nodes prefer to have the same label (or community connections) as their neighbours, this generates communities with the same labels. First Step with Labels: Each node is assigned a distinct label indicating its initial community [26]. Label Propagation: The labels of the nodes are updated by a voting process: in each iteration, each node assigns itself the label that most of its neighbours choose. Tied labels are broken by a tie-breaking rule (e.g., 1rekking, random selection, lowest label). The algorithm runs until no labels are modified or the predefined condition for stopping the algorithm is reached. Imagine that we have a social network in which each person (node) begins with a distinct (initial) label [27]. During the first iteration, everyone examines their neighbours and choose the label that is most commonly used by the group. After several cycles, the groups of people that are more closely connected begin to share the same label, which results in the development of communities inside the social network [28]. Label Propagation is an algorithm utilized for community detection, which operates on the principle of label dissemination across a network [29]. The process is straightforward, efficient, and adaptable to individual preferences. The system's performance may be influenced by the initial labelling and tie-breaking rules applied. Therefore, it is classified as a semi-supervised machine learning algorithm [30].

2.4. Tools, Data Collection, and Description

All tests developed in our work use Python, a general-purpose programming language. Different Python libraries were used for testing, data exploration, splitting datasets, and analysis, such as network analysis, NumPy, and more. This is an analysis of many essential terms and their importance:

- Nodes: Entities inside a network (e.g., individuals, websites, locations).
- Edges: Connections or relationships among the nodes.
- WCC (Weakly Connected Component): A collection of nodes wherein a path exists within any pair of nodes, regardless of the direction of the edges.
- SCC (Strongly Connected Component): A collection of nodes with a directed path existing within every pair of nodes.
- Clustering Coefficient: Assesses the degree of interconnection among a node's neighboring vertices.
- Triangles: Collections of three interconnected nodes.
- Diameter: The maximum length of the shortest path between any two nodes inside the network.
- Effective Diameter: A metric that quantifies the average distance between nodes, taking into account the predominant pairings of nodes.

2.5. Datasets

The analysed network was created by thoroughly investigating the Amazon website and extracting product co-purchasing correlations from the "Customers Who Bought This Item Also Bought" feature. A directed edge was established from product i to product j upon the detection of a statistically significant frequency of co-purchases, indicating that consumers often acquire product j following the purchase of product I [31]. "The Dynamics of Viral Marketing" relates to the notion of viral marketing. It analyses the influence of suggestions from acquaintances and relatives on customer buying behaviour. The researchers J. Leskovec, L. Adamic, and B. Adamic examined a dataset of recommendations from an e-commerce platform [32].

It was determined that just a subset of suggestions results in purchases. Nevertheless, certain recommendations propagate across social networks [4] and result in several sales. The authors found determinants that affect the efficacy of viral marketing initiatives, including the product's price and category. We selected the identical four datasets with varying timestamps for our investigation [33].

- The initial dataset was gathered on March 2, 2003, comprising 262,111 nodes and 1,234,877 edges; the statistics for this dataset are detailed in Table 4a.
- The second dataset was gathered on March 12, 2003, comprising 400,727 nodes and 3,200,440 edges; the statistics for this dataset are detailed in Table 4b.
- The third dataset was gathered on May 5, 2003, comprising 410,236 nodes and 3,356,824 edges; the statistics for this dataset are detailed in Table 4c.
- The final dataset selected was gathered on June 1, 2003, comprising 403,394 nodes and 3,387,388 edges; the statistics for this fourth and final dataset are detailed in Table 4d.

2.6. Data Cleaning

Prior to utilizing these datasets for our testing, we sanitized the data of extraneous information to ensure it yields the necessary findings promptly and without ambiguity. Modularity measures how well a network may be subdivided into clusters (known as modules or clusters) [23]. A relatively high modularity score indicates a dense concentration of connections among nodes within a module, with comparatively few connections between segments. Modularity is extensively employed in optimization techniques to reveal the community structure of a graph or network [33].

The modularity score for a specific network or any graph is generated by “summing number of edges in a cluster and across all clusters and subtracting the expected number of edges that would happen by chance in that cluster” [11]. A partition of the vertices has its own modularity value; larger modularity value indicates a greater connection between the partition and the community structure of the graph.

One such variable when analysing effectiveness is the run time of an algorithm. Because we will be evaluating the computational performance of different algorithms relevant to our research, we will consider run time to be an essential measure of efficacy [9].

2.7. Hypotheses and Implementation

- Hypothesis 1: The dimensions of graphs affect execution time and modularity metrics. We executed four datasets of varying sizes to examine the impact of dataset size on performance and modularity.
- Hypothesis 2: The score of "Label Propagation" should be improved after a number of iterations. To assess this, we allowed more time to run the algorithm. But in the course of our study, we faced label propagation issues. The label propagation started with a random starting value at the start of every epoch, resulting in different outcomes each time.

Measurable outcomes and results from multiple iterations, we found Hypothesis 1 to be a true statement. Each of the algorithms we ran indicated that the size of the associated dataset impacted our runtime and modularity scores. On the other hand, regarding Hypothesis 2, we realized that the results were not reachable, as the propagation with the different labels was always initiated with a new random starting point for each run, and thus, the results varied.

We also pointed out a limited property of the Louvain algorithm: if two subpopulations are labelled the same then cannot connect. The Leiden algorithm was utilized to address this problem.

This method called the Leiden algorithm has some advantages over Louvain — most importantly that it can split clusters instead of just merging them. Its special approach to splitting helps guarantees better connections between clusters. Moreover, we achieved remarkable results in clustering accuracy after several runs. For instance, yet moving one or more from 1 cluster to another does not affect the general quality of the clusters by means of Leiden's Algorithm [10].

What we examined next was Newman's Eigenvector algorithm. Newman's Eigenvector community detection algorithm, which itself hinges on getting the dominant eigenvector from the community matrix. It targets tightly-knit subgraphs through optimized energy calculation of the top non-negative eigenvector spectrum of the modularity matrix. Let the Amazon network G consist of n nodes, and denote it with a n x n adjacency matrix A, so that if and only if there exists an edge among nodes l and m, A(l, m) = 1 else 0.

3. Results

Let's have a look at the outcomes. First, we'll take a look at the execution time; next, we'll check the modularity score.

3.1. Pre-processing and Data Collection

The following graphs show the Runtime of the given five algorithms respectively.

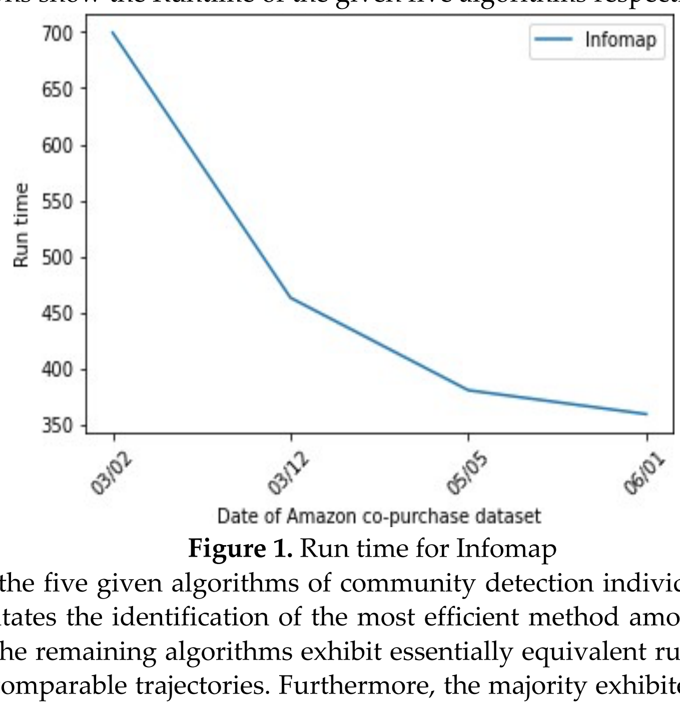

Figure 1. Run time for Infomap

Execution time of the five given algorithms of community detection individually on the given four datasets. Runtime facilitates the identification of the most efficient method among all options. With the exception of Infomap, the remaining algorithms exhibit essentially equivalent runtimes, as evidenced by the graphs displaying comparable trajectories. Furthermore, the majority exhibited the minimal runtime, with Louvain at 5 seconds, Leiden at 6 seconds, Label Propagation at 12 seconds, and Newman at approximately 18 seconds.

Execution time of the five given algorithms of community detection individually on the given four datasets. Runtime facilitates the identification of the most efficient method among all options. With the exception of Infomap, the remaining algorithms exhibit essentially equivalent runtimes, as evidenced by the graphs displaying comparable trajectories. Furthermore, the majority exhibited the minimal runtime, with Louvain at 5 seconds, Leiden at 6 seconds, Label Propagation at 12 seconds, and Newman at approximately 18 seconds.

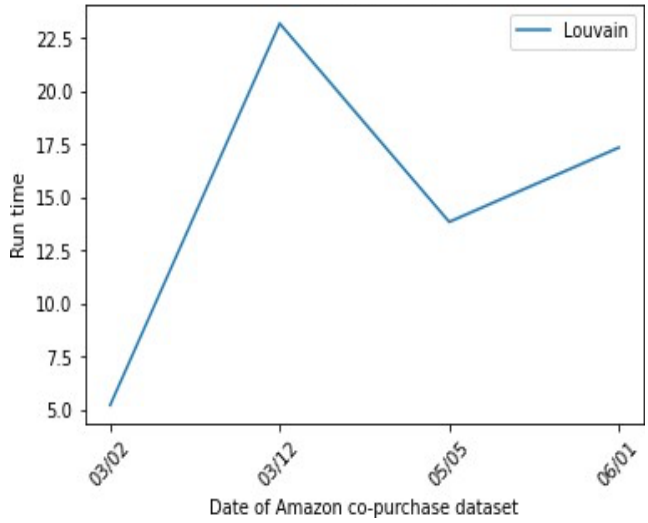

Figure 2. Run time for Louvain

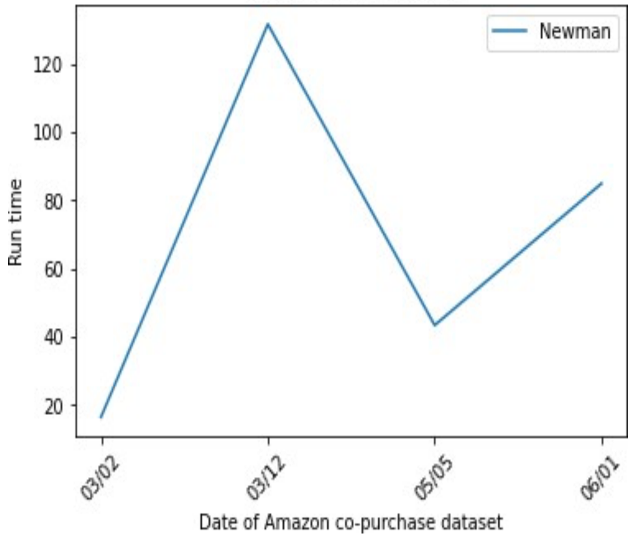

Figure 3. Run time for Newman

Conversely, the Infomap graph had the maximum runtime at the initial point of 03/02, when the number of edges and nodes was minimal; subsequently, the runtime began to decline as the number of edges and nodes rose, illustrated in Figure 3. If you notice timestamp 06/01, Infomap exhibited a minimal runtime of 360 seconds. On 03/12, the graphs by Newman and Louvain, depicted in Figure 4 and Figure 5, had the highest runtimes of 130 seconds and 23 seconds, respectively. The La-bel Propagation graph, Figure 2, dated 05/05, exhibited the maximum runtime of 35 seconds. Figure 5 of the Leiden graph on June 1st exhibited the longest runtime of 14 seconds.

The results demonstrate that Infomap has the lowest efficiency, with an average runtime of 457.64 seconds. Newman concluded that the extended runtime rendered it unsuitable for efficiency discussions. Leiden, Louvain, and Label Propagation show relatively brief runtimes, with Leiden exhibiting the highest efficiency.

4. **Modularity Score**

The following graphs show the **modularity scores** of the respective algorithms executed on each dataset of Amazon.

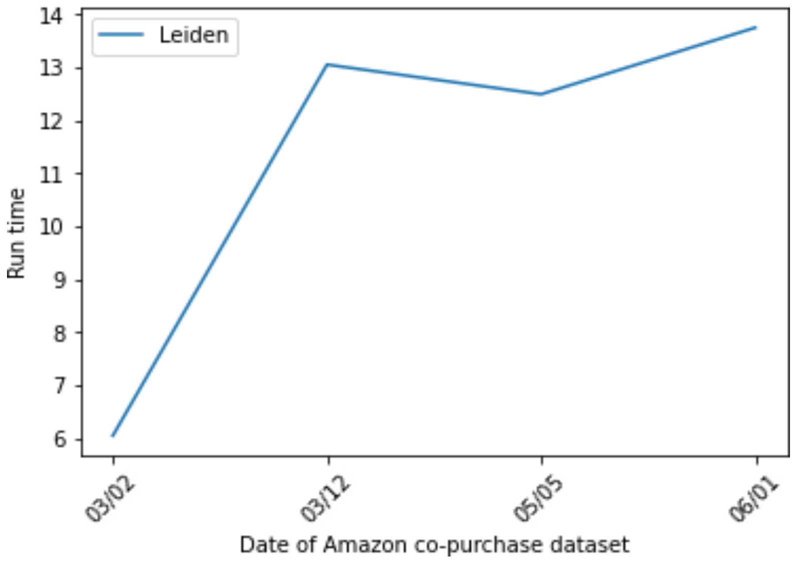

Figure 4. Run time for Leiden

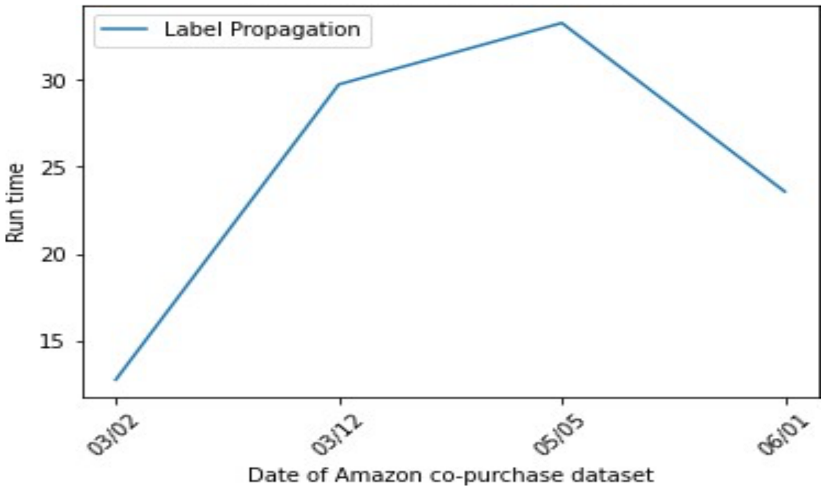

Figure 5. Run time for Label Propagation

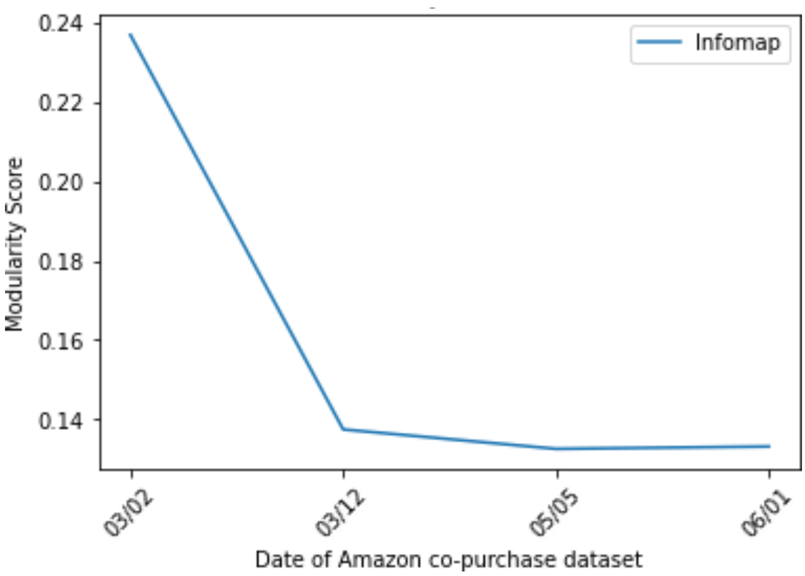

Figure 6. Modularity Score for Infomap

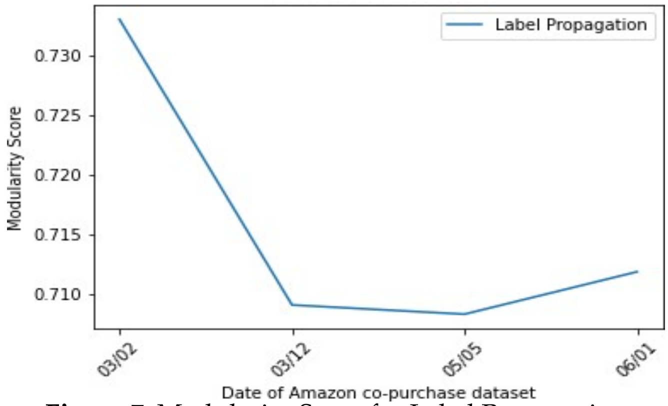

Figure 7. Modularity Score for Label Propagation

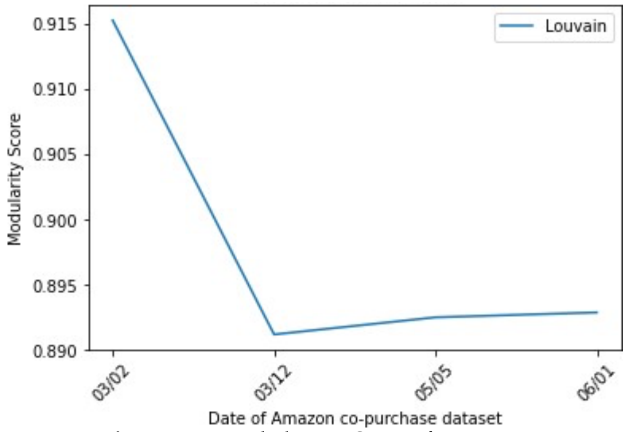

Figure 8. Modularity Score for Louvain

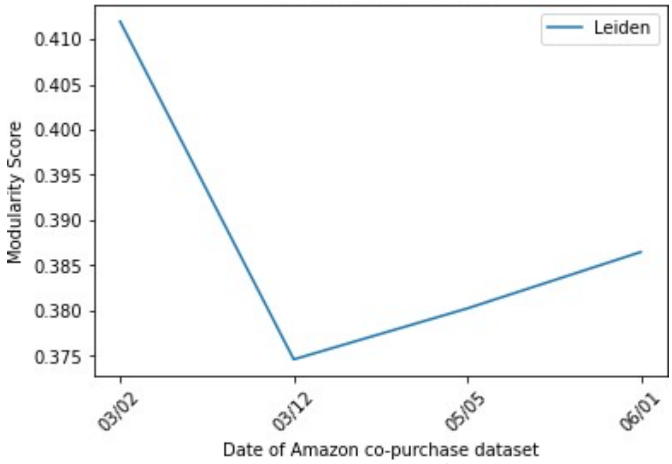

Figure 9. Modularity Score for Leiden

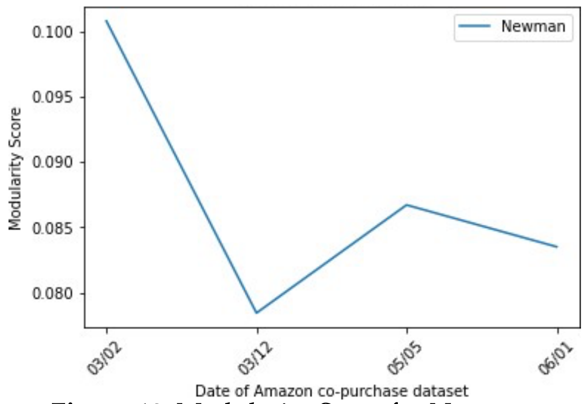

Figure 10. Modularity Score for Newman

All of them produced comparable outcomes. If you notice the initial point 03/02 of the graph, it has the maximum value of modularity score.

- Infomap recorded a peak modularity score of 0.23, while the minimum score was observed on 06/01, at 0.12, as illustrated in Figure 6.
- Label Propagation achieved a maximum modularity score of 0.736, while the minimum score recorded was 0.695 on 05/05. Refer to Figure 7.
- Louvain exhibits a peak modularity score of 0.916, while the lowest recorded score is 0.891, as illustrated in Figure 8.
- Leiden exhibits a peak modularity score of 0.4, while the minimum recorded score is 0.375, as indicated in Figure 9.
- Newman exhibits a peak modularity score of 0.103, while the lowest recorded score is 0.076 at the point 3/12. Figure 10.

The results indicate that Louvain and Label Propagation identified robust connections within the communities of Amazon datasets, while Leiden, Newman, and Infomap revealed relatively weaker connections in the same datasets. Based on our analysis, Louvain and Label Propagation emerge as the most suitable options.

5. Results Discussion

The Infomap and Newman algorithms exhibit high runtime and very low modularity scores, reflecting significant time complexity and very weak community connections. The Leiden algorithm exhibits the shortest runtime; however, its modularity score is relatively unremarkable. Label Propagation exhibits inferior performance compared to Leiden.

The Louvain algorithm balances optimal runtime and high modularity scores, suggesting low time complexity and strong community connections. We can create an effective product recommendation system by combining Louvain with machine learning. Louvain identifies user communities with similar interests, while machine learning predicts preferences and generates personalized user recommendations.

Table 1. Comparisons of algorithms with runtime and modularity score

Average	Newman	Infomap	Leiden	Louvain	Label Propagation
Runtime	69.11	457.74	11.34	14.92	24.84
Modularity Score	0.087	0.16	0.388	0.897	0.716

6. Conclusion

This work aims to create a high-accuracy recommendation system by analyzing two techniques, the Louvain algorithm and machine learning. By this technique, one can be an expert in different domains like movies, books, weather, and so on when analogous data is applied. In addition, K-Means or K-Clusters may work well with Louvain for recommendation and co-purchasing systems as well.

References

1. Rongrong L., Wenzhong, G., Guo, K., and Qiu, Q. "Parallel multi label propagation for overlapping community detection in large scale networks," Multidisciplinary Trends in Artificial Intelligence (MIWAI), Fuzhou, China, 2015, pp. 351-362.
2. Kandel R., and Hiba, B. “A data driven link assessment of Arctic maritime incidents: using machine learning to predict incident types and identify risks factors,” Reliability Engineering and System Safety, 2024, 243, p. 109779
3. Peng, G., Sun, Y., Chen, Q., Li, J., and Liu, z. “The impact of rainfall on urban human mobility from taxi GPS data”, Sustainability, 2022, p. 9355.
4. Jianbin, H., Sun, H., Han, J., and Heng, B. “Density-based shrinkage for revealing hierarchal and overlapping community structure in networks”. Physica A: Statistical Mechanics and its Applications, 2022, vol. 390, no. 11, pp. 2160-2171.
5. Mahmood, T., Ahmed, N., Ahmed, M., & Islam, N. “A Deep Learning Based Approach for Urdu News Summarization”, 2021, arXiv preprint arXiv:2101.12550.
6. Hassan, S., Khan, F., & Hafeez, U. “Clustering-Based Urdu Text Summarization”. In the International Conference on Innovative Computing and Communication Springer, Cham, 2021, pp. 276-286.
7. Khalil, M. I. K.; Afsheen, A.; Taj, A.; Nawaz, A.; Jan, N.; and S. Ahmad, "Enhancing Security Testing Through Evolutionary Techniques: A Novel Model" Journal of Computing & Biomedical Informatics, 2023, vol. 6, no. 1, pp. 375 – 393.
8. Khalil, M. I. K. and Taj, T. “Factors Affecting the Efficacy of Software Development: A Study of Software Houses in Peshawar, Pakistan,” International Review of Basic and Applied Sciences, 2021, vol. 9, no. 3, pp. 385-393.
9. Khalil, M. I. K. Ullah, A.; Taj, A.; Khan, I.A.; Ullah, F.; Taj, F.; and Shah, S. “Analysis of Critical Risk Factors Affecting Software Quality: A Study of KPK, Pakistan Software Industry,” International Review of Basic and Applied Sciences, 2022, vol. 10, no. 2, pp. 338-348.
10. Saqib, A.; Ullah, M.; Hyder, S.; Khatoon, S.; and Khalil, M. I. K. "Creative Decision Making in Leaders: A Case of Beer Game Simulation," Abasyn Journal of Social Sciences, 2020, vol. 12, no. 2, pp. 379-387.
11. Khan, I.A; Ullah, F.; Abrar, M.; Shah, S.; Taj, F. and Khalil, M.I.K. “Ransomware Early Detection Model using API-Calls at Runtime by Random Decision Forests,” International Review of Basic and Applied Sciences, 2022, vol. 10, no. 2, pp. 349-359.
12. Jan, S.; Maqsood, I.; Ahmad, I.; Ashraf, M.; Khan, F. Q.; and Khalil, M. I. K. "A Systematic Feasibility Analysis of User Interfaces for Illiterate Users," Proceedings of the Pakistan Academy of Sciences, 2020, vol. 56, no. 4, pp. 2518-4253.
13. Khalil, M. I. K.; Shah, S. A. A.; Khan, I. A.; Hijji, M.; Shiraz, M.; and Shaheen, Q. “Energy cost minimization using string matching algorithm in geo-distributed data centers,” Computers, Materials, and Continua, 2023, vol. 75, no. 3, pp. 6305-6322.
14. Ahmad, I.; Khalil, M. I. K.; and Shah, S. A. A. "Optimization-based workload distribution in geographically distributed data centers: A survey," International Journal of Communication Systems, 2020, vol. 33, no. 12, p. e4453.
15. Khalil, M. I. K.; Ahmad, I.; Shah, S. A. A.; Jan, S.; and Khan, F. Q. "Energy cost minimization for sustainable cloud computing using option pricing," Sustainable Cities and Society, 2020, vol. 63, p. 102440.
16. Khalil, M. I. K. “Improve quality of service and secure communication in mobile ad hoc networks (MANETS) through group key management,” International Review of Basic and Applied Sciences, 2013, vol. 1, no. 3, pp. 107-115.
17. Muhammad, D.; Ahmad, I.; Khalil, M. I. K.; Khalil, W.; and Ahmad, O. A. “A generalized deep learning approach to seismic activity prediction,” Applied Sciences, MDPI, 2023, vol. 13, p. 1698.
18. Khalil, M. I. K..; Umar, H.; Ullah, K.; Khamosh, S.U., Naqvi, S.F.M., Nawaz, A., and Ahmad, S. “Revolutionizing Schizophrenia diagnosis: A transfer learning approach to accurate classification,” Journal of Computing & Biomedical Informatics, 2024, vol. 07, no. 2, pp. 1-12.
19. Khalil, M. I. K. “Job satisfaction and work morale among PhD’s: A study of public and private sector universities of Peshawar, Pakistan,” International Review of Management and Business Research, 2013, vol. 02, no. 2, p. 362.
20. Ahmad, I.; Ahmad, M. O.; Alqarni, M. A.; Almazroi, A. A.; and Khalil, M. I. K. "Using algorithmic trading to analyze short-term profitability of Bitcoin," PeerJ Computer Science, 2021, vol. 7, p. e337.
21. Khalil, M. I. K.; Shah, S. A. A.; Taj, A.; Shiraz, M.; Alamri, B.; Murawat, S.; and Hafeez, G. “Renewable aware geographical load balancing using option pricing for energy cost minimization in data centers,” Processes, MDPI, 2022, vol. 10, no. 10, p. 1983.

22. Khalil, M. I. K.; Ahmad, A.; Almazroi, A.A. "Energy Efficient Workload Distribution in Geographically Distributed Data Centers," IEEE Access, 2019, vol. 7, no. 1, pp. 82672-82680.
23. Naz, S., and Khalil, M.I.K. "Determinants of job satisfaction: A case study of WAPDA, Peshawar" City University Research Journal, 2012, vol. 2, no. 2, pp. 92-107.
24. Khalil, M. I. K..; Mubeen, A.; Taj, A.; Jan, N.; Ahmad, S. "Renewable and Temperature Aware Load Balancing for Energy Cost Minimization in Data Centers: A Study of BRT, Peshawar," Journal of Computing & Biomedical Informatics, 2023, vol. 06, no. 3, pp. 183-194.
25. Anum, H.; Khalil, M. I. K..; Nawaz, A.; Jan, N.; and Ahmad, S. "Enhancing rumor detection on social media using machine learning and empath features," Journal of Computing & Biomedical Informatics, 2023, vol. 06, no. 2, pp. 272-281.
26. Khalil, M. I. K.; Khan, I.A; Nawaz, A.; Latif, S.; Ahmad, S. and Ahmad, S. "Unveiling the security Maze: A comprehensive review of challenges in Internet of things," Special Issue on Intelligent Computing of Applied Sciences and Emerging Trends (ICASET), Journal of Computing & Biomedical Informatics, 2024, pp. 10-19.
27. Khalil, M.I.K. "Improve quality of service in MANETS through 2-hop routing in cluster-based routing protocol," City University Research Journal, 2012, vol. 2, no. 2, pp. 26-35.
28. Ahmad, I.; Ahmad, M. O.; Alqarni, M. A.; Almazroi, A. A.; and Khalil, M. I. K. "Using algorithmic trading to analyze short-term profitability of Bitcoin," PeerJ Computer Science, 2021, vol. 7, p. e337.
29. Khalil, M. I. K.; Shah, S. A. A.; Taj, A.; Shiraz, M.; Alamri, B.; Murawat, S.; and Hafeez, G. "Renewable aware geographical load balancing using option pricing for energy cost minimization in data centers," Processes, MDPI, 2022, vol. 10, no. 10, p. 1983.
30. Rahman, Z.U..; Khalil, M.I.K..; Nawaz, A.; Khan, I.A..; Jan, N..; and Sheeraz, A. "Analysis and clustering of Pakistani music by lyrics: A study of CokeStudio Pakistan," Journal of Computing & Biomedical Informatics, 2024, vol. 7, no. 1, pp. 281-296.
31. Khan, I..; Khalil, M.I.K..; Nawaz, A.; Khan, I.A..; Zafar, S..; and Sheeraz, A. "Urdu language text summarization using machine learning," Journal of Computing & Biomedical Informatics, 2024, vol. 8, no. 1, pp. 1-10.
32. Akhtar, M.U.; and Khalil, M.I.K. "Link prediction techniques in complex networks," International Review of Basic and Applied Sciences, 2018, vol. 6, no. 8, pp. 60-69.
33. Taj, A., and Khalil, M.I.K. "DDOS defense mechanism and challenges," International Review of Basic and Applied Sciences, 2018, vol. 6, no. 11, pp. 86-93.
34. Khalil, M.I.K., Taj, A., Sadeeq, J. "Selection of cluster head in mobile Ad hoc networks on the basis of battery power," City University Research Journal, 2013, vol. 3, no. 1, pp. 131-139.