Detecting communities on social networks

1st Nguyen Hoang An
University of Information Technology
Ho Chi Minh City, Vietnamese
20521051@gm.uit.edu.vn

***Abstract*—The report discusses a multi-layer clustering algorithm for community detection on social networks, specifically Twitter. This is a community clustering algorithm based on the similarity of structural components and content components, called MCT. From an application perspective, identifying such communities can provide better recommendations, along with other benefits.**

I. Introduction

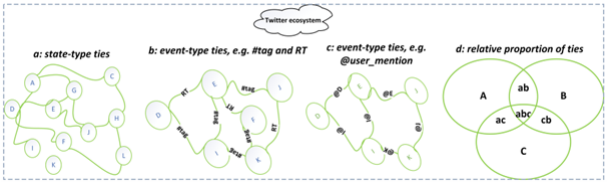

Fig. 1. Types of interaction on Twitter.

Today's social media networks like Twitter and Facebook are communities created by the diverse connections of countless users. Consequently, this ecosystem is highly interconnected and forms various types of interactions, such as interactions among friends of friends, posts with tags, direct mentions @, or retweets on Twitter in Fig. 1. These interactions are what create communities on social media at different levels.

II. Related Work

A. Graph-based and hierarchical methods

Graph-based methods are algorithms that use graphs or network representations to model and analyze the structure of a system. In these methods, nodes represent entities within the system, and edges represent the relationships between those entities. Examples of graph-based methods include centrality measures, community detection algorithms, and network flow analysis.

Hierarchy-based methods are algorithms that organize data or systems into hierarchical structures, where smaller units are nested within larger units. In these methods, the data or system is recursively partitioned into smaller and smaller units based on certain criteria, such as similarity or distance. Examples of hierarchy-based methods include hierarchical clustering algorithms, decision trees, and nested sampling methods.

B. Multiview and bi-modal clustering

Multi-view clustering is an approach to clustering data that involves multiple representations or views of the same underlying dataset. In this approach, each view represents a different aspect or feature of the data, and the goal is to integrate these views to determine a consensus clustering. Multi-view clustering can be useful in cases of complex or high-dimensional data where different views can capture different aspects of the data. Examples of multi-view clustering algorithms include Co-regularized Spectral Clustering and Multi-view K-Means.

Bi-modal clustering is a type of clustering algorithm that operates on data with two distinct modes or sources of information. In this approach, the objective is to identify clusters that are consistent across both modes. Bi-modal clustering can be useful in cases where data has two separate but related sources of information, such as text and images, or gene expression and protein-protein interactions. Examples of bi-modal clustering algorithms include Bi-Clustering and Coupled Bi-Clustering.

III. Datasetss

A. Mental – Health Twitter

This data was collected using the Twitter API. The tweets have been filtered to retain only English contexts. It aims to classify users' mental health. The dataset with fields such as post_id, post_created, post_text, user_id, followers, friends,favourites, statuses, retweets, label. Refer to Fig. 2

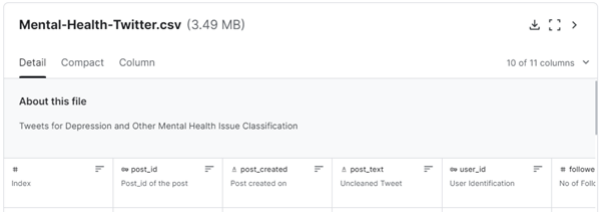

Fig. 2. Mental – Health Twitter datasets

B. Tweets from US Capitol Riots

This data was generated using the Twitter API + Tweepy. It consists of 80,000 tweets from January 6, 2021. The tweets have been stripped of mentions, hyperlinks, emojis, and punctuation. All text has been converted to lowercase. The dataset with fields such as tweet_id, text, query, user_id,

user_name, follower_count, user_tweet_count, likes, retweets, location_name. Refer to Fig. 3

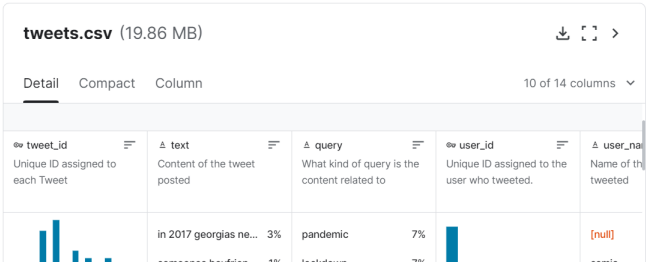

Fig. 3. Tweets from US Capitol Riots datasets

IV. Methodology

The algorithm is defined through 3 stages.

Stage 1: Based on Dyadic Relationships - Bidirectional relationships between two nodes. Simmelian (Transitive) Relationships - Relationships between groups of at least 3 nodes, where there exist direct or indirect relationships between pairs of nodes to calculate the compatibility between node pairs using a algorithm in Fig. 5. The output of the algorithm is an n x n adjacency matrix representing the compatibility between nodes.

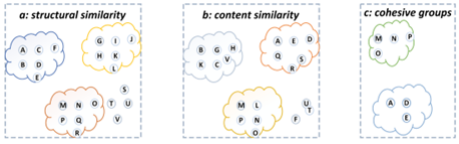

Fig. 4. Node clustering in Detecting communities according to stages

$$p\left(R_{v_i,v_j}\right) = \frac{1}{1 + exp^{\varphi}}$$

Trong đó: $\varphi = -\log\left(\epsilon_{v_i,v_j} + J(_{v_i,v_j})\right) \times \left(\epsilon_{v_i,v_j} + J(_{v_i,v_j})\right)$

Fig. 5. Algorithm for calculating compatibility between Node pairs

Stage 2: We will convert the text components into individual vectors and apply the Latent Dirichlet Allocation (LDA) algorithm to identify a set of topics within the text corpus. The distribution of topics for each tweet stream from related nodes is represented as a distribution of topics in the LDA model. Next, we utilize the Jensen-Shannon distance (JSD) to measure the similarity between two probability distributions. This distance is symmetric and always has a finite value. The result returned is a similarity matrix of size m x m regarding content.

Stage 3: Apply clustering methods on both structural and textual aspects, then perform a function to combine the structural similarity matrix and the content similarity matrix to form an aggregate matrix. Next, transform it into a distance matrix and cluster on this matrix until the stopping criteria are met. The result of the algorithm is the clustered community

V. Experiments

Before proceeding with the main algorithm, we need to evaluate clustering algorithms for structural and content-based clustering.

A. Structural

The execution of the algorithm will produce two sets: Sr for structurally similar nodes and Su for structurally dissimilar nodes. The results are as depicted in the Fig. 6. The structurally similar parts will be saved in the file

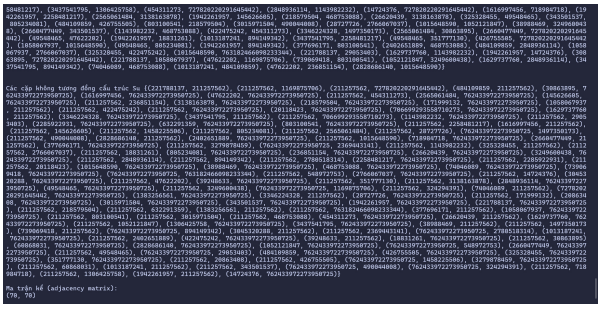

Fig. 6. The results regarding structural similarity

'Sr_data_with_post_text.csv' to serve as input data for the text similarity algorithm.

B. Textual

Similar to the structural similarity algorithm, this algorithm also generates two sets, Tr and Tu, corresponding to the sets of content similar and dissimilar nodes, respectively. The result

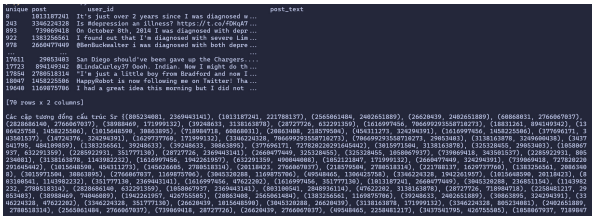

Fig. 7. The results regarding textual similarity

Fig. 7 is clusters of nodes with simple levels of structural and content similarity.

C. Detecting communities

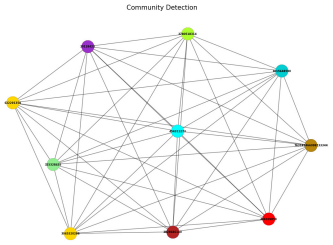

Fig. 8. Clustering model with standard number of communities t = 10

Perform clustering on the network data 'D' with a standard number of communities 't'.

Initialize functions like 'compute_reciprocity_prob', 'compute_text_similarity' , 'f_structural', 'f_textual' , 'compute_pairwise_similarities', 'detecting_communities' to mine communities based on sample data. Due to large data and not powerful personal computers, it leads to very long waiting times when taking large standard community numbers. For that reason, I do it with t = 10.

VI. Result Analysis

The clustering method has demonstrated its ability to detect and identify highly cohesive communities within data. In doing so, it not only helps address challenges stemming from incomplete data but also provides a comprehensive view of the community network. Additionally, analyzing communities at multiple levels aids in a deeper understanding of the relationships and structures within each community. This not only enhances the scalability of the method but also increases its flexibility when applied to various contexts and domains. This has proven to be quite beneficial for intelligent recommendation systems.

VII. Conclusion and Future Work

Regarding the complexity of the proposed method, when the network size is large, the challenge lies in accurately determining when the algorithm for community detection will converge. Therefore, the research team only uses one iteration to analyze the complexity of the algorithm, providing information on future performance. Assuming the execution complexity of a basic algorithm parameterized as f(C), the complexity of the algorithm will be approximately O(f(C) × s × r × m), where s is the number of comparisons to decide the next cluster, r is the size, and m is the number of runs. In terms of execution, the complexity of the algorithm is relatively low; however, it tends to increase as the amount of data increases, thus requiring continued improvement in future work.

The complexity depends on the number of comparisons to decide the next cluster and the number of iterations to determine the optimal number of clusters. So there is a need to research a method that can identify more intrinsic components accurately to minimize the number of comparisons as well as the number of iterations.

References

[1] Isa Inuwa-Dutse, Mark Liptrott, Ioannis Korkontzelos. *A multilevel clustering technique for community detection*. 2021.

[2] Researchgate. *Simmelian Ties on Twitter: Empirical Analysis and Prediction*. 2019.

[3] Researchgate. *Analysis and Prediction of Dyads in Twitter*. 2019.

[4] Infamouscoder, Kaggle. *Mental – Health Twitter datasets*. 2022.