The Truncated Euclidean Hypergraph Average Commute Time Distance-based Clustering Technique

Loc Tran
Department of Electronics, Ho Chi Minh City University of Technology, Vietnam | Vietnam National University Ho Chi Minh City, Vietnam
loctran@hcmut.edu.vn

Kim Anh Phan
Department of Electronics, Ho Chi Minh City University of Technology, Vietnam | Vietnam National University Ho Chi Minh City, Vietnam
pvkanh@hcmut.edu.vn

Hieu Nguyen
Department of Electronics, Ho Chi Minh City University of Technology, Vietnam | Vietnam National University Ho Chi Minh City, Vietnam
hieunt@hcmut.edu.vn

Linh Tran
Department of Electronics, Ho Chi Minh City University of Technology, Vietnam | Vietnam National University Ho Chi Minh City, Vietnam
linhtran@hcmut.edu.vn (corresponding author)

Received: 7 March 2025 | Revised: 27 March 2025 | Accepted: 30 March 2025

 DOI: https://doi.org/10.48084/etasr.10845

ABSTRACT

Clustering is a fundamental problem in data science, with popular approaches including k-means and spectral clustering. However, while k-means is typically limited to tabular datasets and spectral clustering is mainly effective on network data, this work introduces two new methods, the Euclidean hypergraph average commute time distance-based clustering and its truncated variant (i.e., the truncated eigen decomposition technique), which can be applied to any dataset type. Experimental results indicate that both proposed techniques perform at least as well as the conventional Euclidean graph average commute time distance-based clustering and, in some cases, even achieve better results (as measured by the Davies-Bouldin score) because the hypergraph structure captures higher-order relationships among samples. Overall, the proposed methods significantly outperform traditional k-means and spectral clustering when clustering feature vectors.

Keywords-Euclidean hypergraph; clustering techniques; distant-based

I. INTRODUCTION

Clustering plays a vital role in data mining research. It is a technique that partitions samples into clusters such that samples are closer to others in the same clusters than those in others. Its applications are immense, such as mobility pattern clustering [1], text clustering [2-5], image segmentation [6-9], and others. Clustering methods have evolved significantly over the years, with a variety of algorithms developed to uncover intrinsic data structures without relying on predefined labels. The k-means clustering method [1], one of the most widely used partitioning methods, minimizes variance within the cluster but is sensitive to initial centroids and requires specifying the number of clusters in advance. Hierarchical clustering [10], which builds nested clusters either agglomeratively or divisively, offers a dendrogram representation that can provide insights at multiple scales, although its computational complexity can be high for large datasets. Affinity propagation [11] introduces a novel approach by passing messages between data points to automatically determine exemplar clusters, thus eliminating the need to pre-specify the number of clusters, but it may be sensitive to the choice of similarity measures and damping

factors. Other methods, such as density-based clustering methods (e.g., DBSCAN [12]), further extend the clustering toolbox, each with its strengths and weaknesses depending on the data characteristics and the specific application context. These methods are available from the Python sklearn package [13]. However, these techniques/methods can only be applied to tabular datasets. This study uses the k-means clustering technique as the vanilla or baseline technique for three main reasons: First, it is straightforward to implement, second, it scales to large datasets, and third, it guarantees convergence to final solutions.

Other clustering techniques belong to different classes of clustering algorithms and can also be utilized to address the clustering problem but are typically suited for network datasets. A typical technique is spectral clustering, which has shown significant promise in various applications, including mobility pattern analysis [1], and its theoretical foundations have been well documented [14-16]. Another notable approach is the maximum modularity method, which aims to maximize the modularity measure of a graph to find optimal community structures. This method has been effectively applied in several contexts, such as the i-Louvain method for large-scale networks [17], dynamic community detection [18], and user clustering based on preferences in social media platforms such as Twitter [19]. The distributed Louvain algorithm also provides scalable solutions to large-scale community detection problems [20]. These network-specific clustering techniques, including articles based on semantic similarity [21], exploit different principles and methods to identify clusters within complex network data. Furthermore, template-matching approaches for parallelizing the Louvain method [22] and hybrid techniques combining data mining and statistical analysis [23] have been explored to enhance clustering performance.

The Euclidean distance is utilized in the two clustering techniques described above to measure sample similarities. Specifically, in the k-means clustering technique, the Euclidean distances among all samples and all centroids are calculated in the first phase to know which samples belong to which clusters. On the contrary, in the spectral clustering technique, the Euclidean distances among all samples are calculated at the first phase to construct the similarity graph. In the last phase of the spectral clustering technique (i.e., the k-means clustering step), the Euclidean distances are utilized again to measure the similarities among the transformed samples and the centroids.

This study also uses the Euclidean graph average commute time distance-based clustering technique, proposed in [24], to improve clustering performance. In summary, given an undirected weighted graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ with n vertices, the Euclidean graph average commute time distance between two vertices u and v is defined as the expected time it takes the natural random walk starting at vertex u to travel to vertex v and then back to u [25-27]. First, the similarity graph must be constructed from the tabular dataset. Then, the graph Laplacian must be computed. The pseudo-inverse of the graph Laplacian can be computed from the graph Laplacian. Finally, the Euclidean graph average commute time distance can be computed from the pseudo-inverse of the graph Laplacian. The k-medoid clustering technique can address the clustering problem with the Euclidean average commute time distance. This method is generally called the Euclidean graph average commute time distance-based clustering technique.

Since only tabular datasets are given in this work, a graph must be constructed from them. The following section discusses the construction of the graph from the feature vectors. However, there is one weakness associated with the Euclidean graph average commute time distance-based clustering technique. In other words, the assumption of pairwise relationships among the samples in this graph representation is not complete. Consider the case where we would like to partition/segment a set of articles into different topics [28]. The vertices of the graph are the articles. Two articles are connected by an edge (i.e., the relationship) if at least one author is in common. Finally, the clustering technique can be applied to this graph to partition/segment the vertices into groups/clusters. It can be observed that this graph data structure ignores whether one specific author is the author of three or more articles (i.e., the co-occurrence relationship). This method leads to the loss of information. Therefore, the hypergraph data structure is employed for the above relational dataset to overcome this difficulty. In detail, in this hypergraph data structure, the articles are the vertices, and the authors are the hyperedges. A hyperedge can connect more than two vertices (i.e., articles).

This study aimed to extend the work of [24], investigating the development of a novel Euclidean hypergraph average commute time distance-based clustering technique and its variation or not. The contributions of this study include:

- Develops a novel Euclidean hypergraph average commute time distance-based clustering technique.
- Develops a novel Truncated Euclidean hypergraph average commute time distance-based clustering technique.
- Applies these novel clustering techniques to the Zoo, the tiny version of the 20 Newsgroups dataset, and the Citeseer datasets.
- Compares the performance of the Truncated Euclidean hypergraph average commute time distance-based clustering technique with the k-means clustering technique, the spectral clustering technique for feature vectors, the Euclidean graph average commute time distance-based clustering technique, and the Euclidean hypergraph average commute time distance-based clustering technique.

II. EUCLIDEAN GRAPH AVERAGE COMMUTE TIME DISTANCE-BASED CLUSTERING TECHNIQUE

A. Problem Formulation

Suppose we are given a set of samples $\{x_1, x_2, \ldots, x_n\}$, where n is their total number, and the pre-defined number of clusters k. In detail, the adjacency matrix $A \in \mathbb{R}^{n \times n}$ is given, where:

$$A_{ij} = \begin{cases} 1, & \text{if vertex } i \text{ is connected to vertex } j \\ 0, & \text{else} \end{cases} \quad (1)$$

and the feature matrix $X \in \mathbb{R}^{n \times L_1}$ where L_1 is the dimension of the feature vectors. The objective is to output the clusters/groups $A_1, A_2, \dots, A_k$, where $A_i = \{j \mid 1 \leq j \leq n$ and j belongs to cluster $i\}$.

B. Adjacency Matrix Is Not Provided

Suppose we are given the feature matrix $X \in \mathbb{R}^{n \times L_1}$ but not the adjacency matrix $A \in \mathbb{R}^{n \times n}$. The similarity graph can be constructed from these feature vectors using the k-nearest neighbor graph. Sample i is connected with sample j by an edge (no direction: undirected graph) if sample i is among the k nearest neighbors of sample j or sample k is among the k nearest neighbors of sample i. Section IV discusses the construction of the similarity graph from the feature vectors. Finally, the adjacency matrix A representing the similarity graph is obtained.

$$A_{ij} = \begin{cases} 1, & \text{if sample } i \text{ is connected to sample } j \\ 0, & \text{else} \end{cases} \quad (2)$$

Note that this phase is required for spectral clustering and Euclidean graph average commute time distance-based clustering techniques if only the feature matrix is provided.

C. Euclidean Graph Average Commute Time Distance-based Clustering Technique

Currently, the set of feature vectors $\{x_1, x_2, \dots, x_n\}$ and the adjacency matrix are given. Note that $x_i \in \mathbb{R}^{1 \times L_1}$, $1 \leq i \leq n$ and $A \in \mathbb{R}^{n \times n}$. Let D be the diagonal degree matrix of A. In other words, $D_{ii} = \sum_j A_{ij}$. Next, the Laplacian matrix of the graph can be computed as $L = D - A$. Then the pseudo-inverse of the graph Laplacian can be computed as L^+. The naïve definition of the pseudo-inverse of the graph Laplacian is $L^+ = (L^T L)^{-1} L^T$. The average commute time distance can be calculated as follows:

$$d_{ij} = V_G (e_i - e_j)^T L^+ (e_i - e_j) \quad (3)$$

where e_i is the basis vector where the i-th element of e_i is 1 and V_G is the volume of the graph G ($V_G = \sum_{i,j} A_{ij}$). Finally, the clustering problem can be addressed using the k-medoids clustering technique with the Euclidean graph average commute time distance. The k-medoids clustering technique can be found in [29, 30].

III. EUCLIDEAN HYPERGRAPH AVERAGE COMMUTE TIME

A. Problem Formulation

Suppose a set of samples $\{x_1, x_2, \dots, x_n\}$, where n is their total number, and a pre-defined number of clusters k. In detail, the incidence matrix $H \in \mathbb{R}^{n \times n}$ is given, where:

$$H_{ij} = \begin{cases} 1, & \text{if sample } i \text{ belongs to hyperedge } j \\ 0, & \text{else} \end{cases} \quad (4)$$

and the feature matrix $X \in R^{n \times L_1}$, where L_1 is the dimension of the feature vectors. The objective is to output the clusters/groups $A_1, A_2, \dots, A_k$ where $A_i = \{j | 1 \leq j \leq n$ and j belongs to cluster $i\}$.

B. Incidence Matrix of the Hypergraph Is Not Provided

Suppose the feature matrix $X \in \mathbb{R}^{n \times L_1}$ is given but not the incidence matrix $H \in \mathbb{R}^{n \times n}$. The incidence matrix H can be constructed from these feature vectors using the k-nearest neighbor graph. In other words, sample i belongs to hyperedge j if sample i is among the k nearest neighbors of sample j or sample j is among the k nearest neighbors of sample i. Finally, the incidence matrix H is obtained, which represents the hypergraph.

$$H_{ij} = \begin{cases} 1, & \text{if sample } i \text{ belongs to hyperedge } j \\ 0, & \text{else} \end{cases} \quad (5)$$

If only the feature matrix is provided, this phase is required for the Euclidean hypergraph average commute time distance-based clustering technique.

C. Euclidean Hypergraph Average Commute Time Distance-based Clustering Technique

Currently, we have the set of feature vectors $\{x_1, x_2, \dots, x_n\}$ and the incidence matrix H of the hypergraph. Note that $x_i \in \mathbb{R}^{1 \times L_1}$, $1 \leq i \leq n$ and $H \in \mathbb{R}^{n \times m}$, where m is the number of hyperedges in the hypergraph. In some cases, the feature matrix is also the incidence matrix. In these cases, $L_1 = m$.

Let $w(e)$ be the weight of the hyperedge e. Then W is the $R^{m \times m}$ diagonal matrix containing the weights of all hyperedges in its diagonal entries.

From the incidence matrix H and the weight matrix W, the degree of vertex v and the degree of hyperedge e can be defined as follows:

$$d(v) = \sum_{e \in E} w(e) h(v, e) \quad (6)$$

$$d(e) = \sum_{v \in V} h(v, e) \quad (7)$$

Let D_v and D_e be two diagonal matrices containing the degrees of vertices and hyperedges in their diagonal entries, respectively, where D_v is an $R^{n \times n}$ matrix and D_e is an $R^{m \times m}$ matrix. In many clustering applications, if no extra information is available to differentiate the hyperedges, the weight of the hyper-edge $w(e)$ is set to 1. However, if there is a similarity among the vertices that belong to the hyperedge, $w(e)$ can be defined to reflect that. For example, one may set:

$$w(e) = \frac{1}{d(e)} \sum_{(u,v) \subset e} s(u, v) \quad (8)$$

In the above formula, $s(u, v)$ is the similarity between vertex u and vertex v.

In the real hypergraph dataset, some hyperedges may be more important than others and thus will have weights larger than that of others. Hence, this will also lead to the high performance of clustering methods and classification methods associated with the weighted hyperedges of the hypergraph.

Next, the unnormalized Laplacian matrix of the hypergraph can be computed as

$$L_{hyp} = D_v - HWD_e^{-1}H^T \quad (9)$$

Then, the pseudo-inverse of the hypergraph Laplacian can be computed as ${L_{hyp}}^{+}$. The naive definition of the pseudo-inverse of the hypergraph Laplacian is:

$${L_{hyp}}^{+} = \left({L_{hyp}}^{T} L_{hyp}\right)^{-1} {L_{hyp}}^{T} \quad (10)$$

The average commute time distance can be calculated as follows:

$$d_{ij} = V_G \left(e_i - e_j\right)^T {L_{hyp}}^{+} \left(e_i - e_j\right) \quad (11)$$

where e_i is the basis vector where the i-th element of e_i is 1 and V_G is the volume of the hypergraph G. In other words, $V_G = tr(D_v)$. Finally, the clustering problem can be addressed using the k-medoids clustering technique with the Euclidean hypergraph average commute time distance. The k-medoids clustering technique can be found in [29, 30]. This work is illustrated in the following graph.

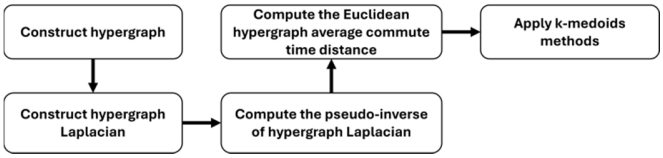

Fig. 1. K-medoids clustering with Euclidean hypergraph average commute time distance.

D. Truncated Euclidean Hypergraph Average Commute Time Distance-based Clustering Technique

The approximation of ${L_{hyp}}^{+}$ is calculated before the Euclidean hypergraph average commute time distance. Suppose ${L_{hyp}}^{+}$ is given. All eigenvalues and eigenvectors of ${L_{hyp}}^{+}$are calculated and sorted in descending order. The first k eigenvectors $v_1, v_2, \dots, v_k$ of ${L_{hyp}}^{+}$ are picked. k can be determined in the following two ways:

- k is a number such that $\frac{\lambda_k}{\lambda_{k+1}}$ is largest for all $1 \le k \le |V| - 1$,
- k is the number such that $\lambda_k - \lambda_{k+1}$ is largest for all $1 \le k \le |V| - 1$.

Let $U \in R^{|V| \times k}$ be the matrix containing the vectors $v_1, v_2, \dots, v_k$ as columns and $\sum \in R^{k \times k}$ be the diagonal matrix containing the k max eigenvalues in the diagonal entries.

Let $L_{hyp}^{approx+} = U \sum U^T$.

The truncated average commute time distance can be calculated as follows:

$$d_{ij}^{truncated} = V_G \left(e_i - e_j\right)^T L_{hyp}^{approx+} \left(e_i - e_j\right) \quad (12)$$

where e_i is the basis vector where the i-th element of e_i is 1 and V_G is the volume of the hypergraph G ($V_G = tr(D_v)$).

Finally, the clustering problem can be addressed using the k-medoids clustering technique with the truncated Euclidean hypergraph average commute time distance.

IV. EXPERIMENTS AND RESULTS

A. Datasets

This study employed the Zoo dataset and the tiny version of the 20 Newsgroups dataset to test the proposed methods. The Zoo dataset [31] is a well-known resource in the field of machine learning and data mining, mainly used for classification and clustering experiments. It contains 101 instances, each representing a different animal, and includes 17 attributes that describe various physical and biological characteristics, such as hair, feathers, eggs, milk, airborne, aquatic, predator, and toothed. The dataset also includes a categorical label that classifies each animal into one of several classes (e.g., mammal, bird, reptile). This dataset is valued for its simplicity and interpretability, making it an excellent tool for exploring feature selection, rule-based classification, and various clustering techniques. In this dataset, each attribute is the hyperedge.

The tiny version of the 20 Newsgroups dataset is a reduced and curated subset of the widely used 20 Newsgroups corpus for text categorization and clustering research [32]. This subset retains the core structure of the original dataset by including documents from 20 distinct newsgroups, each representing different topics such as politics, sports, technology, and science, but with significantly fewer documents per category. The smaller size is especially useful for rapid prototyping and algorithm development, as it allows researchers to test and benchmark their methods on a manageable volume of data without the heavy computational burden of processing the full dataset. Despite its reduced scale, the tiny version preserves the inherent diversity and structure of the full dataset, making it an ideal resource for educational purposes, preliminary experiments in machine learning, and exploring advanced methods in text clustering and classification. The tiny version of the 20 Newsgroups dataset contains binary occurrence data for 7929 words across 387 posts. In this dataset, each word is the hyperedge.

The Citeseer dataset consists of 3,312 scientific publications classified into one of six classes: Agents, AI, DB, IR, ML, and HCI [33]. The citation network consists of 4,732 links. This Citeseer citation network contains 3,312 nodes (i.e., scientific publications) and 4,732 edges (i.e., citation links). Each publication in the Citeseer dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary contains 3,703 unique words. This gives an $R^{3312 \times 3703}$ feature matrix. In this dataset, each word is the hyperedge. Therefore, the feature matrix is also the incidence matrix, as in the previous two datasets [33-37]. Therefore, there is no need to construct the hypergraph from the feature matrix

In the case the adjacency matrices are not given, the similarity graph must be constructed from the feature vectors of the datasets in the following ways:

- The ϵ-neighborhood graph: Connect all the samples whose pairwise distances are smaller than ϵ.
- k-nearest neighbor graph: Sample i is connected to sample j by an edge (no direction: undirected graph) if sample i is

among the k nearest neighbors of sample j or sample j is among the k nearest neighbors of sample i

- The fully connected graph: All samples are connected. In this case, the k-nearest neighbor graph is used to construct the similarity graph from the feature vectors of the Citeseer dataset. Please note that k is chosen to be 5.

B. Experimental Results

The performance of the Truncated Euclidean hypergraph average commute time distance-based clustering technique was compared to those of the Euclidean hypergraph average commute time distance-based clustering technique, the Euclidean graph average commute time distance-based clustering technique, the k-means clustering technique, and the spectral clustering technique for feature datasets. These models were tested using Python on Google Colab with an NVIDIA Tesla K80 GPU and 12 GB RAM.

This Davies-Bouldin score was employed to compare the performance of the clustering techniques. This score is an average similarity measure of each cluster with its most similar one, where similarity is the ratio of within-cluster distances to between-cluster distances. Thus, farther apart and less dispersed clusters result in a better score. The minimum score is zero, This definition shows that the lower the Davies-Bouldin score, the better the clustering results.

Table I shows the performances of the Truncated Euclidean hypergraph average commute time distance-based clustering technique, the k-means clustering technique, the spectral clustering technique for feature vectors, the Euclidean graph average commute time distance-based clustering technique, and the Euclidean hypergraph average commute time distance-based clustering technique on the Zoo dataset. Tables II and III show the performances of these methods on the tiny version of the 20 Newsgroups dataset and the Citeseer dataset.

TABLE I. ZOO DATASET: COMPARISON OF THE CLUSTERING TECHNIQUES

Number of clusters	**2**	**3**	**4**
Truncated Euclidean hypergraph average commute time distance-based clustering	**0.4429**	**0.4359**	**0.5160**
Euclidean hypergraph average commute time distance-based clustering	0.7426	0.7273	0.7181
K-means clustering	1.0035	0.9206	0.8319
Spectral clustering for feature vectors	3.8923	4.4692	4.8032
Euclidean graph average commute time distance-based clustering	**0.4429**	**0.4359**	**0.5160**

TABLE II. TINY VERSION OF THE 20 NEWSGROUPS DATASET: COMPARISON OF THE CLUSTERING TECHNIQUES

Number of clusters	**2**	**3**	**4**
Truncated Euclidean hypergraph average commute time distance-based clustering	**0.9602**	**0.9676**	**0.9645**
Euclidean hypergraph average commute time distance-based clustering	**0.9602**	**0.9602**	**0.9602**
K-means clustering	11.598	11.2901	11.4197
Spectral clustering for feature vectors	2.3318	2.1220	2.5451
Euclidean graph average commute time distance-based clustering	0.9602	0.9669	0.9664

TABLE III. CITESEER DATASET: COMPARISON OF THE CLUSTERING TECHNIQUES

Number of clusters	**2**	**3**	**4**
Truncated Euclidean hypergraph average commute time distance-based clustering	1.3864	1.6787	1.5310
Euclidean hypergraph average commute time distance-based clustering	1.3864	1.6787	1.5310
K-means clustering	3.2186	3.6726	4.7252
Spectral clustering for feature vectors	10.5742	11.3985	10.0772
Euclidean graph average commute time distance-based clustering	**0.6120**	**0.6318**	**0.6429**

These results show that the Truncated Euclidean hypergraph average commute time distance-based clustering technique and the Euclidean hypergraph average commute time distance-based clustering technique are at least as good as the Euclidean graph average commute time distance-based clustering technique, and sometimes have better performance since the hypergraph data structure employs the high order relationships among the samples. This will not lead to loss of information. Moreover, the Truncated Euclidean hypergraph average commute time distance-based clustering technique and the Euclidean hypergraph average commute time distance-based clustering technique are much better than k-means clustering and spectral clustering for feature vectors.

These proposed techniques can be applied not only to these three datasets (i.e., graph or hypergraph datasets) but also to others, such as tabular datasets. This study described how to construct the hypergraph from the tabular dataset in Section III. If samples in the tabular dataset are high-dimensional vectors, dimensionality reduction methods, such as PCA, Laplacian Eigenmaps, etc., can be applied before applying clustering techniques to the tabular dataset. Due to a lack of time and space, such experiments and comparisons were not carried out.

However, there is one weakness associated with the two novel proposed clustering techniques. When new samples arrive, they cannot predict to which clusters these samples belong (unlike the k-means clustering technique). This means that there is a need to update the incidence matrix, to recompute the hypergraph Laplacian, etc. Thus, this technique can only be considered as an off-line clustering technique, although its performance is much higher than the performance of other online clustering techniques (for example, k-means).

V. CONCLUSION

This paper introduced two innovative clustering techniques: the Euclidean hypergraph average commute time distance-based clustering technique and the Truncated Euclidean average commute time distance-based clustering technique. Experimental results demonstrated that these novel methods can effectively utilize high-order relationships within the data, leading to better clustering performance compared to traditional methods. This study showed that hypergraph-based approaches preserve more information and provide superior clustering results by applying them to the Zoo dataset, the tiny version of the 20 Newsgroups dataset, and the Citeseer dataset. Specifically, both the Euclidean hypergraph average commute time distance-based clustering technique and its truncated variant outperformed the k-means and spectral clustering

techniques, highlighting the advantages of incorporating hypergraph data structures in clustering tasks.

Although this work presents improvements in clustering accuracy, it still needs to be completed. Future research will focus on integrating deep learning techniques with the pseudoinverse of the hypergraph Laplacian to further enhance clustering performance. This integration will address limitations and open new avenues to solve complex clustering problems in various domains. Due to lack of time and space, integrating deep learning techniques with the pseudoinverse of the hypergraph Laplacian was not performed, but the reader can check recent works involving combining deep learning techniques with classical clustering techniques [34-36]. The scope of this work involved the development of the classical clustering methods associated with the pseudoinverse of hypergraph Laplacian.

VI. ACKNOWLEDGMENT

The authors acknowledge Ho Chi Minh University of Technology (HCMUT), VNU-HCM, for supporting this study.

REFERENCES

[1] L. H. Tran and L. H. Tran, "Mobility Patterns Based Clustering: A Novel Approach," *International Journal of Machine Learning and Computing*, vol. 8, no. 4, 2018.

[2] J. Yi, Y. Zhang, X. Zhao, and J. Wan, "A Novel Text Clustering Approach Using Deep-Learning Vocabulary Network," *Mathematical Problems in Engineering*, vol. 2017, no. 1, Jan. 2017, Art. no. 8310934, https://doi.org/10.1155/2017/8310934.

[3] A. Hadifar, L. Sterckx, T. Demeester, and C. Develder, "A Self-Training Approach for Short Text Clustering," in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, Florence, Italy, 2019, pp. 194–199, https://doi.org/10.18653/v1/W19-4322.

[4] Y. Liu and X. Wen, "A Short Text Clustering Method Based on Deep Neural Network Model," vol. 29, no. 6, pp. 90–95, 2018.

[5] Z. Dai, K. Li, H. Li, and X. Li, "An Unsupervised Learning Short Text Clustering Method," *Journal of Physics: Conference Series*, vol. 1650, no. 3, Oct. 2020, Art. no. 032090, https://doi.org/10.1088/1742-6596/1650/3/032090.

[6] C. Marcus, "A practical yet meaningful approach to customer segmentation," *Journal of Consumer Marketing*, vol. 15, no. 5, pp. 494–504, Jan. 1998, https://doi.org/10.1108/07363769810235974.

[7] K. K. Tsiptsis and A. Chorianopoulos, *Data Mining Techniques in CRM: Inside Customer Segmentation*. John Wiley & Sons, 2011.

[8] J. Wu and Z. Lin, "Research on customer segmentation model by clustering," in *Proceedings of the 7th International Conference on Electronic Commerce*, May 2005, pp. 316–318, https://doi.org/10.1145/1089551.1089610.

[9] J. J. Jonker, N. Piersma, and D. Van Den Poel, "Joint optimization of customer segmentation and marketing policy to maximize long-term profitability," *Expert Systems with Applications*, vol. 27, no. 2, pp. 159–168, Aug. 2004, https://doi.org/10.1016/j.eswa.2004.01.010.

[10] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," *WIREs Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012, https://doi.org/10.1002/widm.53.

[11] K. Wang, J. Zhang, D. Li, X. Zhang, and T. Guo, "Adaptive Affinity Propagation Clustering." arXiv, May 08, 2008, https://doi.org/10.48550/arXiv.0805.1096.

[12] S. U. Rehman, S. Asghar, S. Fong, and S. Sarasvady, "DBSCAN: Past, present and future," in *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, Bangalore, India, Feb. 2014, pp. 232–238, https://doi.org/10.1109/ICADIWT.2014.6814687.

[13] "2.3. Clustering," *scikit-learn*. https://scikit-learn/stable/modules/clustering.html.

[14] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, Dec. 2007, https://doi.org/10.1007/s11222-007-9033-z.

[15] A. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, 2001, vol. 14, [Online]. Available: https://proceedings.neurips.cc/paper/2001/hash/801272ee79cfde7fa5960571fee36b9b-Abstract.html.

[16] L. H. Tran, L. H. Tran, and H. Trang, "Un-normlized and Random Walk Hypergraph Laplacian Un-supervised Learning," in *Nature of Computation and Communication*, vol. 144, P. C. Vinh, E. Vassev, and M. Hinchey, Eds. Cham: Springer International Publishing, 2015, pp. 254–263.

[17] D. Combe, C. Largeron, M. Géry, and E. Egyed-Zsigmond, "I-Louvain: An Attributed Graph Clustering Method," in *Advances in Intelligent Data Analysis XIV*, 2015, pp. 181–192, https://doi.org/10.1007/978-3-319-24465-5_16.

[18] P. Held, B. Krause, and R. Kruse, "Dynamic Clustering in Social Networks Using Louvain and Infomap Method," in *2016 Third European Network Intelligence Conference (ENIC)*, Wroclaw, Poland, Sep. 2016, pp. 61–68, https://doi.org/10.1109/ENIC.2016.017.

[19] D. L. Sánchez, J. Revuelta, F. De la Prieta, A. B. Gil-González, and C. Dang, "Twitter User Clustering Based on Their Preferences and the Louvain Algorithm," in *Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection*, 2016, pp. 349–356, https://doi.org/10.1007/978-3-319-40159-1_29.

[20] S. Ghosh *et al.*, "Distributed Louvain Algorithm for Graph Community Detection," in *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Vancouver, Canada, May 2018, pp. 885–895, https://doi.org/10.1109/IPDPS.2018.00098.

[21] S. Wang and R. Koopman, "Clustering articles based on semantic similarity," *Scientometrics*, vol. 111, no. 2, pp. 1017–1031, May 2017, https://doi.org/10.1007/s11192-017-2298-x.

[22] S. Bhowmick and S. Srinivasan, "A Template for Parallelizing the Louvain Method for Modularity Maximization," in *Dynamics On and Of Complex Networks, Volume 2: Applications to Time-Varying Dynamical Systems*, A. Mukherjee, M. Choudhury, F. Peruani, N. Ganguly, and B. Mitra, Eds. Springer, 2013, pp. 111–124.

[23] S. Emmons, S. Kobourov, M. Gallant, and K. Börner, "Analysis of Network Clustering Algorithms and Cluster Quality Metrics at Scale," *PLOS ONE*, vol. 11, no. 7, 2016, Art. no. e0159161, https://doi.org/10.1371/journal.pone.0159161.

[24] L. Yen, D. Vanvyve, F. Wouters, F. Fouss, M. Verleysen, and M. Saerens, "Clustering using a random walk based distance measure," in *ESANN'2005 proceedings - European Symposium of Artificial Neural Networks*, Bruges, Belgium, Apr. 2005.

[25] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing Effective Resistance of a Graph," *SIAM Review*, vol. 50, no. 1, pp. 37–66, Jan. 2008, https://doi.org/10.1137/050645452.

[26] M. Duyck and M. Saerens, Distance measures for graph theory: comparisons and analyzes of different methods. .

[27] P. N. Smyrlis, D. C. Tsouros, and M. G. Tsipouras, "Constrained K-Means Classification," *Engineering, Technology & Applied Science Research*, vol. 8, no. 4, pp. 3203–3208, Aug. 2018, https://doi.org/10.48084/etasr.2149.

[28] L. Tran and A. Mai, "Weighted Un-Normalized Hypergraph Laplacian Eigenmaps for Classification Problems," *International Journal of Advance Soft Computing*, vol. 10, no. 3, pp. 190–205, Nov. 2018.

[29] H. S. Park and C. H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, Part 2, pp. 3336–3341, Mar. 2009, https://doi.org/10.1016/j.eswa.2008.01.039.

[30] W. Sheng and X. Liu, "A genetic k-medoids clustering algorithm," *Journal of Heuristics*, vol. 12, no. 6, pp. 447–466, Dec. 2006, https://doi.org/10.1007/s10732-006-7284-z.

[31] R. Forsyth, "Zoo." UCI Machine Learning Repository, 1990, https://doi.org/10.24432/C5R59V.

[32] "20 Newsgroups Data Set." http://qwone.com/~jason/20Newsgroups/.

[33] "CiteSeer for Document Classification," *LINQS*. https://linqs.org/datasets/#citeseer-doc-classification.

[34] Y. Ren *et al.*, "Deep Clustering: A Comprehensive Survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2024, https://doi.org/10.1109/TNNLS.2024.3403155.

[35] M. R. Karim *et al.*, "Deep learning-based clustering approaches for bioinformatics," *Briefings in Bioinformatics*, vol. 22, no. 1, pp. 393–415, Jan. 2021, https://doi.org/10.1093/bib/bbz170.

[36] L. H. Tran, N. Trinh, and L. H. Tran, "Hypergraph convolutional neural network-based clustering technique." arXiv, Sep. 03, 2022, https://doi.org/10.48550/arXiv.2209.01391.

[37] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective Classification in Network Data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, Sep. 2008, https://doi.org/10.1609/aimag.v29i3.2157.

AUTHORS PROFILE

Loc Tran received his B.Sc. and M.Sc. in Computer Science at the University of Minnesota in 2003 and 2012, respectively. Currently, he is a researcher at the John von Neumann Institute in Vietnam. His research interests include spectral hypergraph theory and deep learning.

Kim Anh Phan received his B.Sc. and M.Sc. degrees in Electronics and Telecommunications Engineering from Ho Chi Minh City University of Technology (HCMUT), VNU-HCM (2002, 2012). She is currently a lecturer at the Faculty of Electrical-Electronics Engineering, Ho Chi Minh City University of Technology (HCMUT), VNU-HCM.

Hieu Nguyen received a B.Sc. in control engineering and automation from Ho Chi Minh City University of Technology (HCMUT), Vietnam, in 2019, where he is currently pursuing an M.Sc. in electronics engineering. He is also working as a Lecturer with the Faculty of Electrical-Electronics Engineering, Ho Chi Minh City University of Technology - VNU-HCM.

Linh Tran received a B.Sc. in Electrical and Computer Engineering at the University of Illinois, Urbana – Champaign (2005), and M.Sc. and Ph.D. in Computer Engineering from Portland State University (2006, 2015). Currently, he is a lecturer at the Faculty of Electrical-Electronics Engineering, Ho Chi Minh City University of Technology – VNU-HCM. His research interests include quantum/reversible logic synthesis, computer architecture, hardware-software co-design, efficient algorithms, and hardware design targeting FPGAs and deep learning.