ECAP: Ensemble Clustering using Affinity Propagation

Ankita Sinha
Department of CSE, Rashtrakavi Ramdhari Singh Dinkar College of Engineering, India
rrsdce.ankita@gmail.com(corresponding author)

Rajiv Kumar Ranjan
Department of CSE, Rashtrakavi Ramdhari Singh Dinkar College of Engineering, India
rajivkr1234@gmail.com

Sankalp Sonu
Department of CSE, Rashtrakavi Ramdhari Singh Dinkar College of Engineering, India
sankalpsonuiem@gmail.com

Nitya Nand Jha
Department of Civil Engineering, Rashtrakavi Ramdhari Singh Dinkar College of Engineering, India
nitya.n.jha@gmail.com

Sanjeet Kumar
Department of Civil Engineering, Rashtrakavi Ramdhari Singh Dinkar College of Engineering, India
sanjeetmit08@gmail.com

Received: 27 May 2024 | Revised: 12 June 2024, 27 June 2024, and 13 July 2024 | Accepted: 17 July 2024

ABSTRACT

A vast amount of time-series data is generated from multiple fields. Mining these data can uncover hidden patterns and behavior characteristics. The analysis of such data is complex because they are voluminous and have high dimensions. Clustering can provide a preprocessing step to extract insights. However, clustering such data poses challenges, as many existing algorithms are not efficient enough to handle them. In addition, many traditional and modern clustering algorithms need help with parameter-tuning problems. Ensemble clustering, an amalgamation of clustering algorithms, has emerged as a promising method for improving the accuracy, stability, and robustness of clustering solutions. This study presents Ensemble clustering using Affinity Propagation (ECAP). AP is efficient because it does not require the number of clusters to be specified a priori, allowing the data to reveal its structure. When used in an ensemble framework, the inherent strengths of AP are amplified by integrating multiple clustering results. This aggregation mitigates the influence of any single, potentially suboptimal clustering outcome, leading to more stable and reliable clusters. Extensive experiments were performed on four real-world datasets for rand index, homogeneity, completeness, and V-measure to determine the efficacy of the proposed approach. The results show that the proposed method outperforms AP, Kmeans, and spectral clustering.

Keywords-clustering; affinity propagation; ensemble method

I. INTRODUCTION

Clustering, a fundamental task in unsupervised machine learning, involves partitioning data into groups such that objects within the same group are more similar to each other than those in other groups [1-3]. While effective in many scenarios, traditional clustering algorithms often suffer from limitations such as sensitivity to initialization, dependence on input parameters, and susceptibility to noise. Ensemble clustering has emerged as a powerful approach to mitigate these limitations by combining multiple clustering solutions to achieve more accurate and stable results.

Ensemble algorithms combine multiple clustering solutions to produce a more robust and accurate result [4, 5]. This process typically involves several key steps. At first, the dataset is prepared, which includes preprocessing steps such as handling missing values and scaling features. Next, a diverse set of base clustering algorithms is selected, each capturing different aspects of the data structure. These algorithms are then applied independently to the dataset to generate multiple clustering solutions. The ensemble is formed by combining these solutions using various techniques, such as majority voting or averaging. However, ensemble clustering poses several challenges. Ensuring diversity among base algorithms while avoiding redundancy is crucial, as too much similarity among solutions can undermine the ensemble's effectiveness [6]. Additionally, selecting an appropriate method to combine clustering solutions is challenging, and scalability can be an issue when dealing with large datasets or many base algorithms. Interpreting ensemble results and addressing overfitting are significant concerns, highlighting the need for robust and scalable ensemble clustering techniques.

Several algorithms have been proposed to perform unsupervised classification or clustering, such as hierarchical clustering, density-based clustering, spectral clustering, partitional-based clustering, and so on. Partition-based techniques are widely used among the various methods available due to their easy applicability, simple implementation, and lower time complexity [3]. The k-means algorithm is the flagbearer of all partition-based algorithms. Here, k is the number of clusters that the user must specify before starting the algorithm. The algorithm then selects a set of k initial centers and the remaining points in the dataset are attached to these chosen k centers. The sum of squared error between the cluster centers and the assigned points is minimized iteratively until a suitable set of representatives is found. However, even for $k = 2$, minimizing the objective function is an NP-hard problem, and the algorithm might converge to local minima [1]. In k-means, the quality of the clustering results is also sensitive to the initial seed selection. Furthermore, the k-means algorithm needs help dealing with clusters with different densities or non-spherical shapes. Many improvements over k-means have been proposed to address these issues. One of the most popular enhancements is bisecting k-means [1], an extension of basic k-means. To obtain k clusters in bisecting k-means, the set of points is split into two groups, and then one of the newly formed clusters is further divided. This process continues until k clusters are produced. However, in this case, the user must input the number of clusters, and it is difficult to know the exact number of clusters before starting the algorithm.

The Affinity Propagation (AP) [7] partition-based clustering algorithm overcomes the constraints posed by k-means. The AP clustering algorithm considers each data point a potential cluster center and works based on message passing between the data points. Cluster heads are found from the original data points and exemplars. The algorithm inherently decides the number of clusters, which improves over other partition-based clustering algorithms such as k-means and k-medoids. The algorithm takes $S(i, j)$ as input similarity between the data points i and j, where $i, j \in$ D, where D is the initial dataset. $S(i, j)$ represents how well a data point j can serve as an exemplar for data point i. To automate the number of clusters, the algorithm takes input from an absolute value, $S(i, i)$, called the preference for each data point i. Initially, all data points have the same chance to become an exemplar. Thus, S(i, i) for all data points is set to a standard value at the beginning of the algorithm. The preference value affects the quality of the clustering solution in AP. The value of the input similarity for all pairs of points $(i, j) \in D$ varies between S_{min} and S_{max}, with the median at S_{median}. A smaller preference value S_{min} generates a small number of clusters, whereas S_{max} generates a large number of clusters. Therefore, the value is generally set to S_{median} to develop a moderate number of clusters. One of the significant issues in the AP clustering algorithm is deciding the preference value that generates the final exemplars. The algorithm must be run multiple times using different preference values to develop the optimal number of clusters [2, 7-9]. However, even running the algorithm multiple times does not guarantee optimality. Although the AP clustering algorithm abstains the user from specifying the number of clusters in advance, it, in turn, introduces a new parameter preference that needs to be decided before starting the algorithm.

Ensemble learning [2, 10] originated in supervised learning, with methods such as bagging and boosting gaining popularity in the 1990s. However, the application of ensemble techniques to clustering tasks emerged later. One of the first works in this area is the consensus clustering framework [7]. This study laid the foundation for ensemble clustering by combining multiple clustering solutions to improve overall performance. Since the early 2000s, ensemble clustering has witnessed significant advancements, leading to the development of various techniques and algorithms. Consensus clustering has been a popular approach, seeking a solution that maximizes the agreement among base clusterings. Other techniques include cluster-based ensemble methods, meta-clustering, and weighted clustering ensembles. These methods differ in their underlying principles, combination strategies, and computational complexities, offering various tools to address different clustering challenges. Several studies have contributed to the advancement of ensemble clustering. In [11], valuable insights were provided on the models of consensus and weak partitions. An empirical study on high-dimensional clustering demonstrated the effectiveness of ensemble techniques in challenging scenarios [12]. In [13], supervised and unsupervised learning was combined for text classifications, highlighting the potential of ensemble clustering in real-world applications. In [14], a novel Fast Affinity Propagation clustering approach (FAP) was proposed, which considered both the local and global structural information contained in the datasets. In addition, a new sampling algorithm was proposed to extract a set of representative exemplars. Density-weighted spectral clustering was applied to find the final clustering solution considering the internal data distribution.

Strehl and Ghosh experimentally showed superiority in terms of memory, usage, speed, etc, over the basic AP [7] and spectral clustering [15]. However, in the first stage, while applying AP, the preference value for each point is the same regardless of the distribution of the information contained in the data. MEAP, a multi-exemplar model for AP, is inefficient as a single exemplar for multiclass [16]. The single-exemplar model of AP is extended to multi-model, where each data point is assigned to the most appropriate exemplar, which in turn is assigned to the most suitable super-exemplar to identify the subclass of the category. Max-sum belief propagation was used to solve the NP-hard model, and by using the sparseness of the data, the time complexity was considerably decreased. In [17], AP was applied to incremental problems. Two Incremental Affinity Propagation (IAP) clustering algorithms were proposed, AP clustering based on K-Medoids (IAPKM) and IAP clustering based on Nearest Neighbor Assignment (IAPNA). In K-AP [18], the main concept was to generate a given number of optimal exemplars. However, this approach contradicts the basic advantage of AP, which helps to identify internal data patterns without knowing the exact number of clusters. This study proposes an Ensemble Clustering algorithm using Affinity Propagation (ECAP) to overcome the limitations of existing clustering algorithms. The main contributions of this study are as follows:

- Uses the AP clustering algorithm to diversify the base clustering algorithm.
- Proposes a simple yet powerful consensus function to merge the base clustering results.
- Performs extensive experiments on various real-world datasets for multiple cluster validity indices to determine the efficacy of the proposed method.

II. PROPOSED WORK

A. *Affinity Propagation Clustering*

Affinity Propagation (AP) [19] is a sophisticated and versatile clustering algorithm that has gained prominence due to its unique approach and several inherent advantages over traditional clustering methods. AP stands out for its ability to dynamically determine the number of clusters and its effectiveness in handling large datasets. AP identifies exemplars among data points and forms clusters based on these exemplars. Unlike k-means clustering, which requires the number of clusters to be predetermined, AP does not need this information upfront. This feature is particularly useful when the optimal number of clusters is still being determined. The process begins by measuring the similarity between pairs of data points. This similarity is the negative squared Euclidean distance, although other similarity measures can be used depending on the context. AP then uses an iterative process to exchange messages between data points. These messages fall into responsibility (r_{ik}) and availability (a_{ik}). r_{ik} indicates how suitable a point is to be the exemplar for point i. It is updated based on the similarity between the points and the availability messages received from other points. a_{ik} reflects how appropriate it would be for point i to choose point k as its exemplar, considering the support from other points. The availability is updated based on the responsibilities received from other points. These messages are iteratively updated until they converge, indicating that the clustering structure has stabilized. The final clusters are formed by assigning each point to the exemplar, which maximizes the sum of the responsibility and availability messages.

B. *Ensemble Clustering*

Ensemble algorithms [1, 20, 21] combine the output of several clustering algorithms or runs of the same algorithm to produce a final clustering solution. Figure 1 shows how ensemble clustering works. It is also called consensus clustering, which combines multiple base clustering algorithm results, which are finally combined using the consensus function. Ensemble clustering aggregates the outputs of multiple individual clustering algorithms or runs of the same algorithm to improve the quality and stability of clustering results. Initially, diverse base clusters are generated by applying various clustering techniques or varying parameters within a single algorithm. These base clusterings are then merged into an ensemble matrix or partition, where each element represents the membership of a data point across different clusters. Techniques such as voting, averaging, or consensus combine these memberships into a unified representation. Finally, a final clustering solution is derived from this combined representation, often by applying another algorithm to refine the ensemble clustering result. Ensemble clustering leverages the diversity between base clusterings to mitigate the limitations of individual algorithms, resulting in more robust and reliable clusterings that better capture the underlying structure of the data.

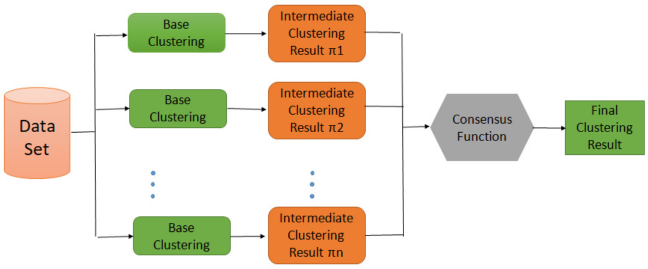
Data Set
Base Clustering
Base Clustering
Base Clustering
Intermediate Clustering Result π1
Intermediate Clustering Result π2
Intermediate Clustering Result πn
Consensus Function
Final Clustering Result

Fig. 1. Ensemble clustering.

C. *Problem Statement*

Let $X = x_1, x_2, ..., x_n$ be a dataset containing n points, each having d attributes. Each $x_i \in X$ is represented as (x_{i1},x_{i2},..., x_{id}). Let $\pi = \pi_1, \pi_2, \ldots, \pi_M$ be the M outputs generated by the base clusters, known as the ensemble members. π_r represents the set of all clusters generated by the base algorithm, π_r=C_1^r,C_2^r,.......$C_k r^r$, such that $U_{j=1}^{kr} C_j^r = X$. Here k_r is the number of clusters generated by $r_r h$ clustering. The problem is to find the final set of clusters denoted by $\pi_r = C_1^*, C_2^*, ..., C_k^*$, where k is the final number of clusters and n^* denotes the final set of clusters. To solve this problem, this study proposes a novel algorithm called ECAP, as shown in Algorithm 1.

D. *Base Clustering Using AP*

A homogeneous ensemble is applied in the proposed ECAP algorithm. This approach applies a single algorithm to the dataset, changing the parameter values in each base algorithm. The AP algorithm is used, which has two parameters, the preference and the damping factor, that affect the final result. Preference governs the number of clusters. Thus, the quality of the clustering solution and damping factor prevent the algorithm from getting stuck in local minima. It is a competitive learning algorithm that offers an effective approach to clustering by adopting a set of prototype vectors to represent patterns in input data.

ALGORITHM 1: ECAP

```
Input:
  $p_i$← the preferences of the AP algorithm
  $\lambda_i$← the threshold of the AP algorithm
  itr← the number of iterations fixed by the user
Output:
  K ← top clusters

for i = 1 to itr do
  Run AP($p_1$, $\lambda_1$)
  if (number of clusters ≥ $\sqrt{n}$)
    continue
  store result in $\pi_i$;
end for
remove duplicate clusters
remove clusters with 0.5 deviation from mean
find rank $\forall ci \in \pi_i$
K = mode($k_i$), where $k_i$ is clusters get by ensemble members
select top K rank clusters
return K
```

E. Consensus Function

After getting intermediate results, the next motive is to create a consensus among the intermediate solutions to find the final result. ECAP employs majority voting to find the final result from the base clustering output. Voting is performed based on the following criteria:

- Removal of duplicates with an error range.
- Ranking cluster heads.
- Number of cluster heads generated.

Repeated cluster heads are removed. If two cluster heads are very close, one is eliminated from the solution list. The two centers are assumed to be close based on the mean of the dataset. If two or more centers fall in the range of ±5% from each other, then they are eliminated. Cluster heads are ranked based on the cardinality of repetition in the base clustering solution. For example, if the center c_k is repeated p times in the M cluster ensemble, then the rank of c_k will be p. The second step of voting is applied to the generated clusters. Here, the mode of centers is voted as the final number of clusters, and the top k-ranking centers are selected as the final cluster heads. Figure 2 shows the consensus function.

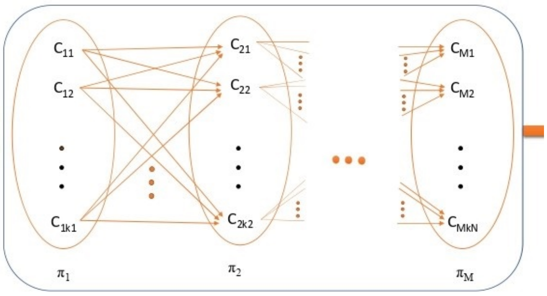

Fig. 2. Consensus function.

III. EXPERIMENTAL RESULTS AND DISCUSSION

A series of experiments on real-world and synthetic data was performed to evaluate the performance of the proposed algorithm. The experiments were performed on a PC with an Intel Core i7-1255U 1.70 GHz CPU, 16GB RAM, on Windows 11. The programs were written in Python. Table I shows the details of the datasets obtained [22].

TABLE I. DATASET DESCRIPTION

Dataset	Number of points	Number of dimensions
Haberman	306	3
Iris	150	4
Wifi	2000	7
Ionosphere	351	34

A. Performance Metrics

1) Rand Index

Rand Index (RI) is a statistical technique to measure similarity between two clusters [20, 23]. It requires knowledge of true class assignments. As shown in Figure 3, the RI of the proposed algorithm is close to k-means, spectral clustering, and AP for Iris, Ionosphere, Haberman, and Wifi datasets.

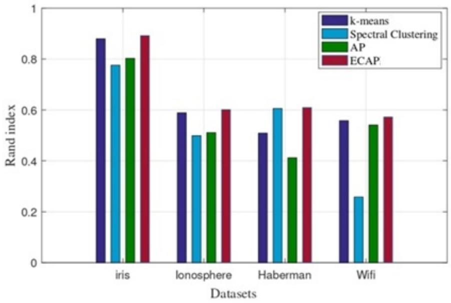

Fig. 3. Rand index.

2) Homogeneity

Homogeneity measures the similarity of two samples by assuming that only members of one class are present in each cluster [23]. Homogeneity is calculated using the following formula, where $H(\frac{C}{K})$ is the conditional entropy of the class and $H(C)$ is the entropy of the class.

$$h = 1 - \frac{H(\frac{C}{K})}{H(C)} \tag{1}$$

Figure 4 shows that the clusters are composed of data points that are highly consistent within their respective classes.

3) Completeness

Contrary to homogeneity, completeness assumes that a class's members are all assigned to the same cluster. Completeness is calculated using

$$c = 1 - \frac{H(\frac{K}{C})}{H(C)} \tag{2}$$

Figure 5 shows that the completeness of the proposed algorithm is close to k-means, spectral clustering, and AP for the Iris, Ionosphere, Haberman, and Wifi datasets.

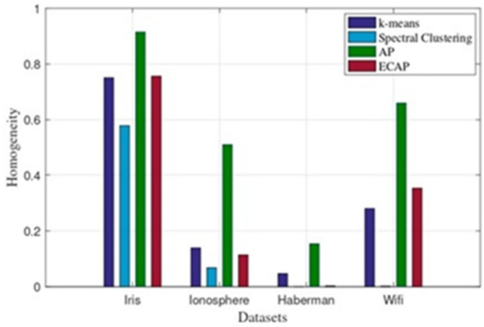

Fig. 4. Homogeneity.

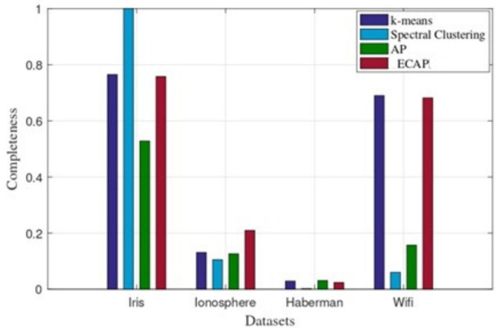

Fig. 5. Completeness.

4) V-measure

V-measure is the harmonic mean of homogeneity and completeness [23], defined as shown in (3). The default value of β is 1. The asymmetric score can assess the consistency of two disjoint assignments on the same dataset.

$$v = \frac{(1+\beta) \times h \times c}{\beta \times (h + c)} \quad (3)$$

As shown in Figure 6, the v-measure of the proposed algorithm is close to k-means, spectral clustering, and AP for the Iris, Ionosphere, Haberman, and Wifi datasets.

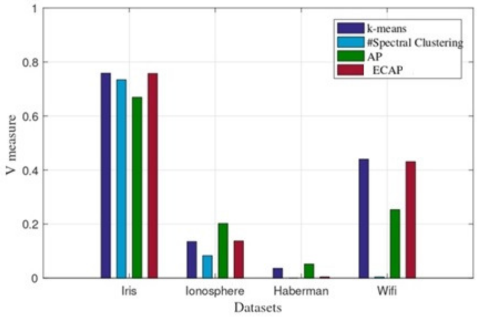

Fig. 6. V-measure.

The above metrics indicate that the proposed ECAP method outperformed the other algorithms in terms of performance.

IV. CONCLUSION AND FUTURE WORK

This study presented Ensemble Clustering using Affinity Propagation (ECAP) to leverage the strengths of AP, such as its ability to handle non-convex clusters and identify exemplar points in an ensemble framework. The advantages of AP are amplified through the integration of multiple clustering results. This aggregation mitigated the influence of a single suboptimal clustering outcome and resulted in a more stable and reliable clustering solution. The results of ECAP were compared with AP, k-means, and spectral clustering for four real-world datasets using four cluster validity indices to show the efficacy of ECAP over existing clustering algorithms.

REFERENCES

[1] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, Jun. 2005, https://doi.org/10.1109/TPAMI.2005.113.

[2] Z. Xu, Y. Lu, and Y. Jiang, "Research on Mini-Batch Affinity Propagation Clustering Algorithm," in *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, Shenzhen, China, Jul. 2022, pp. 1–10, https://doi.org/10.1109/DSAA54385.2022.10032450.

[3] A. E. Ezugwu *et al.*, "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," *Engineering Applications of Artificial Intelligence*, vol. 110, Apr. 2022, Art. no. 104743, https://doi.org/10.1016/j.engappai.2022.104743.

[4] A. S. Alkarim, A. S. Al-Malaise Al-Ghamdi, and M. Ragab, "Ensemble Learning-based Algorithms for Traffic Flow Prediction in Smart Traffic Systems," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13090–13094, Apr. 2024, https://doi.org/10.48084/etasr.6767.

[5] T. Boongoen and N. Iam-On, "Cluster ensembles: A survey of approaches with recent extensions and applications," *Computer Science Review*, vol. 28, pp. 1–25, May 2018, https://doi.org/10.1016/j.cosrev.2018.01.003.

[6] W. Ismaiel, A. Alhalangy, A. O. Y. Mohamed, and A. I. A. Musa, "Deep Learning, Ensemble and Supervised Machine Learning for Arabic Speech Emotion Recognition," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13757–13764, Apr. 2024, https://doi.org/10.48084/etasr.7134.

[7] A. Strehl and J. Ghosh, "Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2002.

[8] H. Ge, L. Wang, H. Pan, Y. Zhu, X. Zhao, and M. Liu, "Affinity Propagation Based on Structural Similarity Index and Local Outlier Factor for Hyperspectral Image Clustering," *Remote Sensing*, vol. 14, no. 5, Jan. 2022, Art. no. 1195, https://doi.org/10.3390/rs14051195.

[9] J. Liu, G. Liao, J. Xu, S. Zhu, C. Zeng, and F. H. Juwono, "Unsupervised Affinity Propagation Clustering Based Clutter Suppression and Target Detection Algorithm for Non-Side-Looking Airborne Radar," *Remote Sensing*, vol. 15, no. 8, Jan. 2023, Art. no. 2077, https://doi.org/10.3390/rs15082077.

[10] W. Alsabhan, "Human–Computer Interaction with a Real-Time Speech Emotion Recognition with Ensembling Techniques 1D Convolution Neural Network and Attention," *Sensors*, vol. 23, no. 3, Jan. 2023, Art. no. 1386, https://doi.org/10.3390/s23031386.

[11] A. Topchy, A. K. Jain, and W. Punch, "Clustering ensembles: models of consensus and weak partitions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1866–1881, Sep. 2005, https://doi.org/10.1109/TPAMI.2005.237.

[12] X. Z. Fern, and C. E. Brodley, "Cluster ensembles for high dimensional clustering : an empirical study," *Journal of Machine Learning Research*, vol. 5, pp. 155–175, 2004.

[13] F. Maturo and R. Verde, "Combining unsupervised and supervised learning techniques for enhancing the performance of functional data classifiers," *Computational Statistics*, vol. 39, no. 1, pp. 239–270, Feb. 2024, https://doi.org/10.1007/s00180-022-01259-8.

[14] F. Shang, L. C. Jiao, J. Shi, F. Wang, and M. Gong, "Fast affinity propagation clustering: A multilevel approach," *Pattern Recognition*, vol. 45, no. 1, pp. 474–486, Jan. 2012, https://doi.org/10.1016/j.patcog.2011.04.032.

[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.

[16] C. D. Wang, J. H. Lai, C. Y. Suen, and J. Y. Zhu, "Multi-Exemplar Affinity Propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2223–2237, 2013, https://doi.org/10.1109/TPAMI.2013.28.

[17] L. Sun and C. Guo, "Incremental Affinity Propagation Clustering Based on Message Passing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 11, pp. 2731–2744, 2014, https://doi.org/10.1109/TKDE.2014.2310215.

[18] X. Zhang, W. Wang, K. Nørvåg, and M. Sebag, "K-AP: Generating Specified K Clusters by Efficient Affinity Propagation," in *2010 IEEE International Conference on Data Mining*, Sydney, Australia, Sep. 2010, pp. 1187–1192, https://doi.org/10.1109/ICDM.2010.107.

[19] D. Dueck, "Affinity Propagation: Clustering Data by Passing Messages," Ph.D. dissertation, University of Toronto, Canada, 2009.

[20] J. Han, J. Pei, and H. Tong, *Data Mining: Concepts and Techniques*, 4th edition. Cambridge, MA, USA: Morgan Kaufmann, 2022.

[21] A. Miltiadous *et al.*, "An Ensemble Method for EEG-based Texture Discrimination during Open Eyes Active Touch," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12676–12687, Feb. 2024, https://doi.org/10.48084/etasr.6455.

[22] "UCI Machine Learning Repository." http://archive.ics.uci.edu/.

[23] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognition*, vol. 46, no. 1, pp. 243–256, Jan. 2013, https://doi.org/10.1016/j.patcog.2012.07.021.