ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10890745

Weighted Density for The Win: Accurate Subspace Density Clustering

Maixuan Peng[1,‡], Yuyang Wu[2,1,‡], Yang Lu[3], Mengke Li[4], Yiqun Zhang[1,5,*], Yiu-ming Cheung[5]
[1]School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China
[2]Faculty of Science, Wilfrid Laurier University, Brantford, Canada
[3]Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, Xiamen, China
[4]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
[5]Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China

***Abstract*—k-clustering typically struggles with the detection of irregular-distributed clusters due to the natural bias, while density clustering usually cannot well-adapt to different datasets and clustering tasks as it is not an oriented optimization process. This paper, therefore, proposes to perform density clustering in dynamically learned subspaces. To exploit the irregular-distributed clusters obtained by density clustering for the subspace determination, we design a new strategy to appropriately evaluate the importance of attributes. It turns out that the proposed Weighted Density-based Subspace Clustering (WDSC) algorithm inherits the unbiased merits of density clustering, and also upgrades the unlearning density clustering to be learnable under the subspace learning paradigm of k-clustering. A comprehensive evaluation including significance tests, ablation studies, qualitative comparisons, etc., shows the superiority of WDSC.**

***Index Terms*—Subspace clustering, density-based clustering, attributes weighting, unsupervised learning**

I. Introduction

Cluster analysis is essential for uncovering hidden patterns and is widely applied to pattern recognition, streaming data analysis, multi-view clustering, and so on [1]–[4]. k-clustering and density-based clustering are two representative types of partitional clustering adopting very different partition strategies and generating clusters with divergent distributions. However, they both rely on the prior knowledge of the dataset [5]. That is, k-clustering requires the proper number k of spherical-shaped clusters as input, while density clustering prefers banded clusters. Nevertheless, exact prior knowledge is usually unavailable for real datasets with various types of cluster distributions, which makes the selection of the proper type of clustering method a non-trivial task [6]–[8].

More specifically, k-clustering divides the dataset into k disjoint clusters according to the similarity between data objects and cluster descriptors, e.g., centers, modes, prototypes, etc. Typical algorithms include k-means [9] and its variants [10]–[12]. The recent advanced k-clustering that consider subspace cluster effect [13], [14], streaming data [15]–[18], unknown k [19]–[22], mixed-attributes [23]–[26], etc. can also be found in the literature. From the perspective of cluster distribution detection, k-clustering generally works well in exploring convex and spherical clusters formed by gathering samples with cluster centers. However, such bias restricts k-clustering from being competent in forming irregular-shaped clusters.

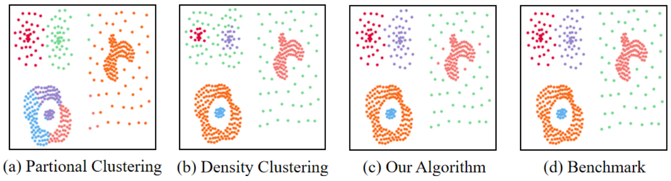


Fig. 1: Comparison of different types of clustering.

By contrast, density clustering tends to define densely connected distribution areas as clusters, avoiding introducing an obvious cluster shape bias. DBSCAN [27], a representative algorithm, does not require k to be given and can effectively handle noisy data but is computationally expensive in high dimensions. Its improved version named BLOCK-DBSCAN [28] introduces an approximation mechanism of the density measurement to reduce computational complexity. Unlike DBSCAN, DPC [29] is another classical type of density clustering that focuses on determining the cluster centers by density peaks. Based on DPC, SNNDPC [30] further enhances local density estimation and better identifies complex clustering structures using shared nearest neighbors. Meanwhile, DenMune [31] enhances clustering quality by automatically detecting and excluding noisy points. Nevertheless, they all share a common limitation of treating attributes equally [32]. Moreover, they do not involve a learning process to fit certain clustering objectives, which limits their adaptability [33].

As shown in Fig. 1.(a), partition clustering struggles with non-spherical or distribution-connected clusters. Although density clustering relieves bias toward specific cluster shapes as seen in Fig. 1.(b), it involves laborious computation and tends to explore clusters with uniform density levels. Therefore, an ideal clustering algorithm should be able to accommodate non-spherical clusters and flexibly adapt to complex cluster distributions with various densities. To achieve such flexible cluster distribution exploration, subspace clustering that finely explores clusters in corresponding subspaces formed by the weighted attributes [34] is an ideal solution. However, most subspace clustering algorithms are built upon the k-clustering

[‡]Co-first author, *Corresponding author: yqzhang@gdut.edu.cn

paradigm, and thus the irregular-shaped clusters obtained by density clustering are unsuitable for evaluating the importance of attributes in forming subspaces.

This paper, therefore, proposes the Weighted Density-based Subspace Clustering (WDSC) algorithm to perform density clustering in dynamically learned subspaces. As shown in Fig. 1.(c), WDSC can more flexibly explore arbitrary-shaped clusters in different density levels in the subspaces. More specifically, we introduce an innovative spherical re-clustering mechanism to re-partition each cluster formed by the density clustering into compact spherical-shaped clusters preferred by the k-clustering for more accurate attribute weight evaluation. An iterative optimization strategy is also designed to allow automatic updates of attribute weights based on the current clusters during learning. Experimental results show that WDSC significantly outperforms the advanced partition, density, and subspace clustering methods in terms of clustering accuracy. The main contributions of this paper are three-fold:

- To achieve unbiased and adaptive clustering, we propose a subspace clustering paradigm that integrates density and k-clustering merits, which pioneers subspace density cluster learning and serves as a general framework for enhancing the existing density-based clustering methods.
- Aiming at the issue of evaluating attribute importance according to the formed irregular clusters by density clustering, we develop a re-clustering method to subdivide the density clusters for more appropriate weight evaluation, thereby appropriately determining the subspace.
- The proposed method inherits the unbiasedness of density clustering and applies to a wide range of cluster distribution types. The adaptability of the clustering process to the task at hand is also further enhanced by the iterative optimization of the subspace selection. As a result, WDSC demonstrates higher stability and accuracy than state-of-the-art clustering methods.

II. Proposed Method

In this section, we introduce Weighted Density-based Subspace Clustering (WDSC), which integrates attribute weight learning into density clustering to optimize the clustering process within a dynamically learned subspace. This section comprises three parts: 1) How to perform density clustering with a given subspace; 2) How the attribute weights for forming subspace are determined through spherical reclustering; and 3) How to perform iterative optimization to learn the data objects partition and subspaces simultaneously.

A. Subspace Density Clustering

Assuming a given weight vector $\mathbf{w} = [w_1, w_2, \ldots, w_m]$ corresponding to m attributes of a dataset $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ containing n objects, this subsection discusses how to perform density clustering based on fixed $\mathbf{w}$. Each object $\mathbf{x}_i$ is represented as a vector $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{im}]$ in an m-dimensional attribute space. For the i-th object, its K-nearest neighbors are denoted as $N_K(\mathbf{x}_i)$. Note that we use K to indicate the number of neighbors and k for the number of clusters. The Shared Nearest Neighbors (SNN) [30] of two objects are the common objects in their neighbor sets:

$$SNN(\mathbf{x}_i, \mathbf{x}_j) = N_K(\mathbf{x}_i) \cap N_K(\mathbf{x}_j). \tag{1}$$

To reflect the varying importance of attributes, we introduce the weighted distance to allow greater contribution of significant attributes:

$$D_{ij} = \sqrt{\sum_{l=1}^{m} w_l \cdot (x_{il} - x_{jl})^2}, \tag{2}$$

where w_l represents the weight of the l-th attribute, x_{il} and x_{jl} are the values of $\mathbf{x}_i$ and $\mathbf{x}_j$ on the l-th attribute, respectively. Based on SNN and D_{ij}, the weighted SNN similarity $S(\mathbf{x}_i, \mathbf{x}_j)$ of two objects can be calculated as:

$$S(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \dfrac{|SNN(\mathbf{x}_i, \mathbf{x}_j)|^2}{\sum\limits_{p} \left(D_{ip} + D_{jp}\right)}, & \text{if } SNN(\mathbf{x}_i, \mathbf{x}_j) \neq \emptyset \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $p \in SNN(\mathbf{x}_i, \mathbf{x}_j)$. This measures the connection strength between two objects by taking both neighborhood overlaps and attribute importance into account.

The weighted local density P_i of $\mathbf{x}_i$ is defined as the sum of $S(\mathbf{x}_i, \mathbf{x}_j)$ between $\mathbf{x}_i$ and its k-most similar neighbors $N_k(\mathbf{x}_i)$ calculated by $S(\mathbf{x}_i, \mathbf{x}_j)$:

$$P_i = \sum_{\mathbf{x}_j \in N_k(\mathbf{x}_i)} S(\mathbf{x}_i, \mathbf{x}_j). \tag{4}$$

Here, the k-most similar neighbors $N_k(\mathbf{x}_i)$ of $\mathbf{x}_i$ are the k points with the highest similarity values to $\mathbf{x}_i$, where the similarity is measured by according to Eq. (3).

Since P_i is incapable of reflecting the local isolation of $\mathbf{x}_i$ compared to the other high-density objects, we further define the weighted distance Λ_i from object $\mathbf{x}_i$ to its nearest larger density object $\mathbf{x}_j$ to distinguish objects that can be adopted as prominent cluster centers:

$$\Lambda_i = \min_{\mathbf{x}_j : P_j > P_i} \left[D_{ij} \cdot \left(\sum_{p \in N_K(\mathbf{x}_i)} D_{ip} + \sum_{q \in N_K(\mathbf{x}_j)} D_{jq} \right) \right]. \tag{5}$$

Among all $\mathbf{x}_j$ that satisfy $P_j > P_i$, the one with the smallest value calculated by Eq. (5) determines the value of Λ_i. To identify cluster centers, we define the overall decision value Γ_i for $\mathbf{x}_i$ by comprehensively considering P_i and Λ_i as:

$$\Gamma_i = P_i \times \Lambda_i. \tag{6}$$

Intuitively, an object with a larger Γ value is more likely to be a prominent cluster center as it has a higher local density and is relatively far away from the other high-density samples (i.e., the other cluster center candidates). After determining k cluster centers according to Γ, the weighted similarity defined by Eq. (3) is utilized to compute the object-cluster similarities to assign each sample to its most similar cluster center [30].

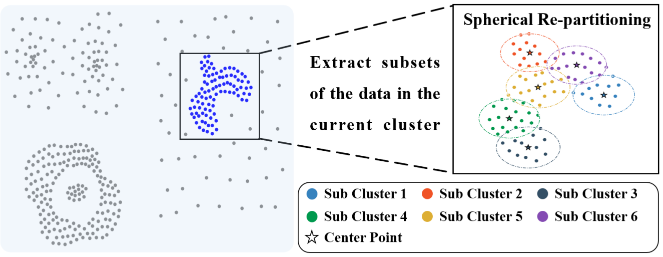


Fig. 2: Intracluster spherical re-partition.

B. *Weights Updating Based on Cluster Re-partitioning*

The previous section discussed how to perform density clustering with the given weight vector $\mathbf{w}$. This section will discuss how to compute $\mathbf{w}$ during the clustering process.

Suppose the dataset has been partitioned into k density clusters with irregular shapes as described by Section II-A. If we directly use these clusters to evaluate the importance of attributes according to the cluster compactness reflected by the intra-cluster dissimilarity using Euclidean distance, unsatisfactory results will occur. For instance, a prominent ring-like or line-like cluster may still result in a very high intra-cluster distance. On the other hand, if we use the similarity defined by Eq. (3), the distribution direction of non-spherical clusters will over-dominate the attribute importance, which makes the attribute weighting meaningless to clustering.

Accordingly, as shown in Fig. 2, we first re-partition an obtained cluster C_r into k_r sub-clusters $\{C_{r1}, C_{r2}, \ldots, C_{rk_r}\}$ using k-medoids [35], where k_r is the number of clusters at the time when the Silhouette Coefficient (SC) [36] is closest to its maximum (i.e., 1) determined after multiple partition attempts. Then we use sub-clusters and their corresponding medoids $\{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{kr}\}$ to compute the weights of all the m attributes denoted as a vector $\mathbf{w}_r = [w_{r1}, w_{r2}, \ldots, w_{rm}]$. To compute the weight value (e.g., w_{rl}), we first compute the within-cluster variance v_{rl} of the l-th attribute by:

$$v_{rl} = \sum_{b=1}^{k_r} \sum_{\mathbf{x}_b \in C_{rb}} (x_{bl} - c_{bl})^2, \tag{7}$$

where x_{bl} and c_{bl} are the values of $\mathbf{x}_b$ and $\mathbf{c}_b$ on the l-th attribute, respectively. Then the weight w_{rl} of the l-th attribute reflected by cluster C_r is computed by normalizing the within-cluster variance across all attributes. Since the importance is inversely related to the variance, w_{rl} can be quantified as:

$$w_{rl} = \left(\sum_{t=1}^{m} \left(\frac{v_{rl}}{v_{rt}} \right)^{\frac{1}{\beta - 1}} \right)^{-1}. \tag{8}$$

Here, the hyper-parameter β controls the sensitivity of the weighting mechanism to attribute variance [34].

To mitigate interference from noise or outlier clusters, global attributes-weight vector $\mathbf{w}$ is obtained by weighting each $\mathbf{w}_r$ according to the size of each cluster $|C_r|$ by:

$$\mathbf{w} = \frac{1}{\sum_{i=1}^{k} |C_r|} \sum_{r=1}^{k} \Big(|C_r| \cdot \mathbf{w}_r \Big). \tag{9}$$

Algorithm 1 WDSC: Weighted Density-based Subspace Clustering

Input: Dataset X, initial $\mathbf{w}$, hyperparameter σ.
Output: Clustering results $C = \{C_1, C_2, \ldots, C_k\}$.

```
repeat
    Calculate $S$, $P$, $\Lambda$ and $\Gamma$ according to Eqs. (3)-(6);
    Select $k$ cluster centers according to $\Gamma$;
    Partition $X$ into $k$ clusters, obtain $C = \{C_1, C_2, \ldots, C_k\}$;
    Partition each $C_r$ into $k_r$ clusters where $r = \{1, 2, ..., k\}$;
    Compute attribute weights $\mathbf{w}_r$s by Eqs. (7) and (8);
    Update $\mathbf{w}$ by combining $\mathbf{w}_r$s according to Eq. (9);
    Update cluster assignments using the new weight vector $\mathbf{w}$;
    Evaluate clustering quality by SC;
    Calculate acceptance probability based on $T$ by Eq. 10;
    Update $T$ according to the number of iterations by $T = 1/g$;
until $T$ falls below threshold $\sigma$ for five consecutive iterations.
```

TABLE I: Statistics of the UCI (left part) and synthetic (right part) datasets. n, m, and k denote the number of samples, dimensions, and ground-truth clusters, respectively.

ID	Datasets	$<n, m, k>$	ID	Datasets	$<n, m, k>$
1	Sonar	<208, 60, 2>	7	Pathbased	<300, 2, 3>
2	Wine	<178, 13, 3>	8	Aggregation	<788, 2, 7>
3	Iris	<150, 4, 3>	9	Jain	<373, 2, 2>
4	Heart	<303, 13, 2>	10	Spiral	<312, 2, 3>
5	Pima	<768, 8, 2>	11	Moon	<210, 2, 2>
6	Leaf	<340, 15, 30>	12	CMC	<1002, 2, 3>

So far, we have obtained the overall attribute weights $\mathbf{w}$ that facilitate subspace for the following density clustering.

C. *Learning Algorithm*

To enhance both accuracy and robustness, we adopt an iterative updating strategy inspired by simulated annealing to search for more appropriate C and $\mathbf{w}$. At each iteration g, the algorithm generates a new clustering solution and evaluates its quality using SC metric with a higher value indicating better cluster quality. If the new score exceeds the current best, the algorithm accepts and updates the best solution. Otherwise, acceptance occurs obey probability AP, which is defined as:

$$AP = \exp\Big(\frac{|SC^{\text{new}} - SC^{\text{old}}|}{T}\Big), \tag{10}$$

where T is a parameter that decreases over iterations (i.e., $T = 1/g$). Early in the process, the learning algorithm enables the occasional acceptance of suboptimal solutions to avoid premature convergence to local optima. As g increases, T decreases, making the algorithm more likely to reject inferior solutions, and ultimately leading to a stable and effective clustering configuration. The process terminates automatically when T falls below a predefined threshold σ for five consecutive iterations, ensuring timely convergence and reducing the risk of overfitting. The overall learning algorithm is summarized as the Algorithm 1.

III. Experiments

Seven counterparts: including two classical (i.e., W-KMeans (WKM) [34] and DBSCAN [27]), two representative density-based (i.e., SNNDPC [30] and DenMune [31]), and three SOTA (i.e., DPC-CE [37], VDPC [38] and ICKDP [39])

TABLE II: Clustering performance in terms of AC (left) and the results of ablation study (right). The best performance on each dataset is marked in boldface. The average ranking of each method on all the datasets is reported in the last row.

Data	**Comparative results**								**Ablation study**		
	W-KM	**DBSCAN**	**SNNDPC**	**DenMune**	**DPC-CE**	**VDPC**	**ICKDP**	**WDSC**	**Baseline**	**WDC**	**WDSC**
Pathbased	0.6133	0.6500	0.9767	0.9533	0.7333	**1.0000**	0.7017	0.9900	0.9767	0.9867	**0.9900**
Aggregation	0.6320	0.8198	0.9848	0.9924	0.9987	**1.0000**	0.9987	0.9962	0.9848	**0.9962**	**0.9962**
Jain	0.8820	0.5389	0.8606	**1.0000**	**1.0000**	**1.0000**	0.9196	**1.0000**	0.8606	0.8901	**1.0000**
Spiral	0.3462	0.6250	**1.0000**	0.7580	**1.0000**	**1.0000**	0.4359	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Moon	0.9524	0.0762	**1.0000**	**1.0000**	0.6905	0.8095	0.8000	**1.0000**	**1.0000**	**1.0000**	**1.0000**
CMC	0.4571	0.7715	**1.0000**	**1.0000**	0.6776	0.4721	0.6826	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Sonar	0.5048	0.5385	0.6202	0.5337	0.5865	0.5385	0.6109	**0.7115**	0.6202	0.6635	**0.7115**
Wine	0.9600	0.5169	0.9663	0.9607	0.7079	0.5674	0.7079	**0.9888**	0.9663	0.9775	**0.9888**
Iris	0.9067	0.7867	0.9773	0.9000	0.8533	0.8800	0.9000	**0.9800**	0.9733	**0.9800**	**0.9800**
Heart	0.7162	0.3795	0.6964	0.4719	0.7096	0.7096	0.6964	**0.8317**	0.6964	0.8185	**0.8317**
Pima	0.5508	0.6602	0.6484	0.6458	0.6471	0.5771	0.5678	**0.6745**	0.6484	0.6575	**0.6745**
Leaf	**0.4529**	0.0853	0.3941	0.1765	0.1353	0.1147	0.1382	0.3745	0.3941	**0.4000**	0.3745
Avg. Rank	5.67	6.08	3.00	4.00	4.33	4.25	5.08	1.58	2.42	1.50	1.17

clustering algorithms are chosen. The hyperparameter of the proposed WDSC is set at $\sigma = 0.0001$. Since the optimal K values for selecting neighbors of each data object in terms of distance can be determined by their corresponding SC performance, they are not treated as hyperparameters and are automatically searched during clustering.

12 Datasets: are collected from [40]–[44] and the statistics are shown in Table I. The attributes of datasets are pre-processed by using the min-max normalization [30] to have a uniform value range [0,1] before the experiments.

Validity Indices: Clustering ACcuracy (AC) [45] is chosen for evaluating the clustering performance. Larger values indicate better clustering performance.

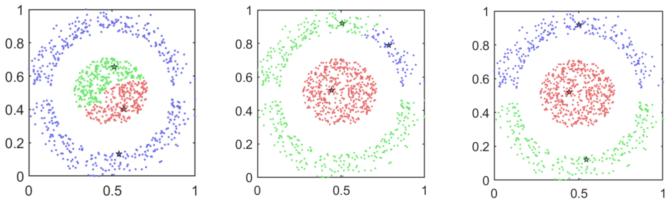
Comparative Results: All the results are averaged on 10 runs of the experiments. The bold values indicate the best performance for a dataset. It can be seen in Table II that the proposed WDSC consistently outperforms the other algorithms on most datasets, and reaches an average ranking of 1.5, highlighting its superiority and reliability on different datasets.

Ablation Study: Key components of WDSC, i.e., weights computation and subspace learning, are validated by comparing the baseline (i.e., SNNDPC) and the attribute-weighted version of WDSC without iterative subspace learning (denoted as WDC). As shown in Table II, WDC outperforms Baseline in most cases, proving the efficacy of the weights computation. WDSC performs better than WDC on most datasets, confirming the effectiveness of the learning mechanism.

Intuitive Study: The Learning process of WDSC is illustrated in Fig. 3. The algorithm begins with a rough clustering configuration shown by Fig. 3 (a), where the cluster centers are not accurately positioned. As the algorithm proceeds, adjustments are made to optimize the cluster centers in the learned subspace, see Fig. 3 (b). The clustering converges where the cluster centers and boundaries are well-recognized demonstrated by Fig. 3 (c).

IV. Concluding Remarks

A novel clustering approach called WDSC that combines unbiased cluster distribution search with subspace learning has been proposed. Built upon density-based clustering, WDSC effectively detects any-shaped clusters and incorporates an attribute weights learning mechanism. It turns out that WDSC can more finely explore clusters without being restricted to: 1) Gaussian distribution assumption of k-clustering, and 2) Fixed subspace with identical importance of attributes. As a result, WDSC achieves superior clustering performance and robustness compared to the SOTAs. Since this work focuses on static numerical datasets with a fixed k, future extensions could target dynamic or streaming datasets with mixed attributes, incorporating automatic determination of the optimal k.



(a) Initialization (b) Process (c) Convergence

Fig. 3: Clustering procedures of the proposed WDSC.

V. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grants: 62476063, 62102097, 62376233, and 62306181, the NSFC/Research Grants Council (RGC) Joint Research Scheme under grant: N_HKBU214/21, the Natural Science Foundation of Guangdong Province under grants: 2024A1313010039, 2024A1515010163, and 2023A1515012855, the Natural Science Foundation of Fujian Province under grant: 2024J09001, the General Research Fund of RGC under grants: 12201321, 12202622, and 12201323, the RGC Senior Research Fellow Scheme under grant: SRFS2324-2S02, the Shenzhen Science and Technology Program under grant: RCBS20231211090659101, the National Key Laboratory of Radar Signal Processing under grant: JKW202403, and the Xiaomi Young Talents Program.

References

[1] Haize Hu, Jianxun Liu, Xiangping Zhang, and Mengge Fang, "An effective and adaptable k-means algorithm for big data cluster analysis," *Pattern Recognition*, vol. 139, pp. 109404, 2023.

[2] Chao Zhang, Deng Xu, Xiuyi Jia, Chunlin Chen, and Huaxiong Li, "Continual multi-view clustering with consistent anchor guidance," in *IJCAI*, 2024, pp. 5434–5442.

[3] Deng Xu, Chao Zhang, Zechao Li, Chunlin Chen, and Huaxiong Li, "Fast disentangled slim tensor learning for multi-view clustering," *ArXiv Preprint ArXiv:2411.07685*, 2024.

[4] Miroslava Nedyalkova, Sergio Madurga, and Vasil Simeonov, "Combinatorial k-means clustering as a machine learning tool applied to diabetes mellitus type 2," *International Journal of Environmental Research and Public Health*, vol. 18, no. 4, pp. 1919, 2021.

[5] Yiu-ming Cheung and Yiqun Zhang, "Fast and accurate hierarchical clustering based on growing multilayer topology training," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 876–890, 2018.

[6] Yunfan Zhang, Xiaopeng Luo, Qingsheng Chen, Rong Zou, Yiqun Zhang, and Yiu-ming Cheung, "Towards unbiased minimal cluster analysis of categorical-and-numerical attribute data," in *ICPR*, 2024, pp. 254–269.

[7] Sen Feng, Mingjie Zhao, Zhanpei Huang, Yuzhu Ji, Yiqun Zhang, and Yiu-ming Cheung, "Robust qualitative data clustering via learnable multi-metric space fusion," in *ICASSP*, 2025, pp. 1–5.

[8] Yunfan Zhang, Rong Zou, Yiqun Zhang, Yue Zhang, Yiu-ming Cheung, and Kangshun Li, "Adaptive micro partition and hierarchical merging for accurate mixed data clustering," *Complex & Intelligent Systems*, vol. 11, no. 1, pp. 1–14, 2025.

[9] Kristina P. Sinaga and Miin-Shen Yang, "Unsupervised k-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020.

[10] Anton V. Ushakov and Igor Vasilyev, "Near-optimal large-scale k-medoids clustering," *Information Sciences*, vol. 545, pp. 344–362, 2021.

[11] Feng Jiang, Guozhu Liu, Junwei Du, and Yuefei Sui, "Initialization of k-modes clustering using outlier detection techniques," *Information Sciences*, vol. 332, pp. 167–183, 2016.

[12] Jinchao Ji, Tian Bai, Chunguang Zhou, Chao Ma, and Zhe Wang, "An improved k-prototypes clustering algorithm for mixed numeric and categorical data," *Neurocomputing*, vol. 120, pp. 590–596, 2013.

[13] Yuan Fang, Geping Yang, Xiang Chen, Zhiguo Gong, Yiyang Yang, Can Chen, and Zhifeng Hao, "Landmark-based k-factorization multi-view subspace clustering," *Information Sciences*, vol. 667, pp. 120480, 2024.

[14] Ken Chen, Yong Tang, Long Wei, Pengfei Wang, Yong Liu, and Zhongming Jin, "Sparse subspace clustering for stream data," *IEEE Access*, vol. 9, pp. 57271–57279, 2021.

[15] Mingjie Zhao, Yiqun Zhang, Yuzhu Ji, and Yang Lu, "Unsupervised concept drift detection via imbalanced cluster discriminator learning," in *PRCV*, 2023, pp. 31–43.

[16] Yunfan Zhang, Yiqun Zhang, Yang Lu, Mengke Li, Xi Chen, and Yiu-ming Cheung, "Asynchronous federated clustering with unknown number of clusters," *arXiv preprint arXiv:2412.20341*, 2024.

[17] Xiaopeng Luo, Yiqun Zhang, Yuzhu Ji, Peng Liu, and Yiu-ming Cheung, "Efficient topology-driven clustering for imbalanced streaming biomedical data analysis," in *BIBM*, 2024, pp. 1–6.

[18] Yiqun Zhang, Pengkai Wang, Sen Feng, Zexi Tan, Xiaopeng Luo, Yuzhu Ji, Rong Zou, and Yiu-Ming Cheung, "Learning self-growth maps for fast and accurate imbalanced streaming data clustering," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2025.

[19] Abhimanyu Kumar, Abhishek Kumar, Rammohan Mallipeddi, and Dong-Gyu Lee, "High-density cluster core-based k-means clustering with an unknown number of clusters," *Applied Soft Computing*, vol. 155, pp. 111419, 2024.

[20] Shenghong Cai, Yiqun Zhang, Xiaopeng Luo, Yiu-Ming Cheung, Hong Jia, and Peng Liu, "Robust categorical data clustering guided by multi-granular competitive learning," in *ICDCS*, 2024, pp. 288–299.

[21] Yue Zhang, Xinfa Liao, Qingsheng Chen, Haotian Wu, and Yiqun Zhang, "Fedhc: Learning imbalanced clusters via federated hierarchical clustering," in *PRCV*, 2024, pp. 505–521.

[22] Rong Zou, Yunfan Zhang, Yiqun Zhang, Yang Lu, Mengke Li, and Yiu-ming Cheung, "Federated clustering with unknown number of clusters," in *DOCS*, 2024, pp. 671–677.

[23] Junyang Chen, Yuzhu Ji, Rong Zou, Yiqun Zhang, and Yiu-ming Cheung, "Qgrl: quaternion graph representation learning for heterogeneous feature data clustering," in *KDD*, 2024, pp. 297–306.

[24] Yiqun Zhang and Yiu-Ming Cheung, "Graph-based dissimilarity measurement for cluster analysis of any-type-attributed data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 6530–6544, 2022.

[25] Yiqun Zhang and Yiu-Ming Cheung, "A new distance metric exploiting heterogeneous interattribute relationship for ordinal-and-nominal-attribute data clustering," *IEEE Transactions on Cybernetics*, vol. 52, no. 2, pp. 758–771, 2020.

[26] Yiqun Zhang, Yiu-Ming Cheung, and Kay Chen Tan, "A unified entropy-based distance metric for ordinal-and-nominal-attribute data clustering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 1, pp. 39–52, 2019.

[27] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*, 1996, vol. 96, pp. 226–231.

[28] Yewang Chen, Lida Zhou, Nizar Bouguila, Cheng Wang, Yi Chen, and Jixiang Du, "Block-dbscan: Fast clustering for large scale data," *Pattern Recognition*, vol. 109, pp. 107624, 2021.

[29] Alex Rodriguez and Alessandro Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[30] Rui Liu, Hong Wang, and Xiaomei Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *Information Sciences*, vol. 450, pp. 200–226, 2018.

[31] Mohamed Abbas, Adel El-Zoghabi, and Amin Shoukry, "Denmune: Density peak based clustering using mutual nearest neighbors," *Pattern Recognition*, vol. 109, pp. 107589, 2021.

[32] Yiqun Zhang, Yiu-ming Cheung, and An Zeng, "Het2hom: Representation of heterogeneous attributes into homogeneous concept spaces for categorical-and-numerical-attribute data clustering," in *IJCAI*, 2022, pp. 3758–3765.

[33] Yiqun Zhang and Yiu-ming Cheung, "Learnable weighting of intra-attribute distances for categorical data clustering with nominal and ordinal attributes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3560–3576, 2021.

[34] Joshua Zhexue Huang, Michael K. Ng, Hongqiang Rong, and Zichen Li, "Automated variable weighting in k-means type clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 657–668, 2005.

[35] Donghua Yu, Guojun Liu, Maozu Guo, and Xiaoyan Liu, "An improved k-medoids algorithm based on step increasing and optimizing medoids," *Expert Systems with Applications*, vol. 92, pp. 464–473, 2018.

[36] Adil M Bagirov, Ramiz M Aliguliyev, and Nargiz Sultanova, "Finding compact and well-separated clusters: Clustering using silhouette coefficients," *Pattern Recognition*, vol. 135, pp. 109144, 2023.

[37] Wenjie Guo, Wenhai Wang, Shunping Zhao, Yunlong Niu, Zeyin Zhang, and Xinggao Liu, "Density peak clustering with connectivity estimation," *Knowledge-Based Systems*, vol. 243, pp. 108501, 2022.

[38] Yizhang Wang, Di Wang, You Zhou, Xiaofeng Zhang, and Chai Quek, "Vdpc: Variational density peak clustering algorithm," *Information Sciences*, vol. 621, pp. 627–651, 2023.

[39] Wenjie Guo, Wei Chen, and Xinggao Liu, "Density peak clustering by local centers and improved connectivity kernel," *Information Sciences*, vol. 666, pp. 120439, 2024.

[40] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas, "Clustering aggregation," *Acm Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, pp. 4–es, 2007.

[41] Hong Chang and Dit-Yan Yeung, "Robust path-based spectral clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 191–203, 2008.

[42] Anil K. Jain and Martin HC Law, "Data clustering: A user's dilemma," in *PReMI*. Springer, 2005, pp. 1–10.

[43] Markelle Kelly, Rachel Longjohn, and Kolby Nottingham, "The uci machine learning repository," `https://archive. ics. uci. edu`, 2023.

[44] Yizhang Wang, Jiaxin Qian, Muhammad Hassan, Xinyu Zhang, Tao Zhang, Chao Yang, Xingxing Zhou, and Fengjin Jia, "Density peak clustering algorithms: A review on the decade 2014–2023," *Expert Systems with Applications*, vol. 238, pp. 121860, 2024.

[45] Xiaofei He, Deng Cai, and Partha Niyogi, "Laplacian score for feature selection," in *NeurIPS*, 2005, vol. 18.