A Comparative Survey of Clustering Algorithm

He Zhi, 21215122
Department of computer science and engineering
Sun Yat-Sen University

Abstract—Clustering is an algorithm widely used in data mining. In reality, the category labels of many data may be unknown. To reveal the properties and internal laws contained in the data, certain data analysis techniques are required, which are generally called "unsupervised learning" techniques. Clustering is an important technology among them. It is the most studied and widely used, and many algorithms are derived. Each algorithm is divided into various forms to be suitable for different scenarios. By comparing these clustering algorithms, in this paper, we discuss the utility of various clustering algorithms, so that readers can understand their advantages and disadvantages in specific scenarios.

Index Terms—clustering, data mining, unsupervised learning

1. INTRODUCTION

Clustering is an important unsupervised learning technique, which can divide the data set without category label into several disjoint subsets, so that the data samples are located in different "clusters", these "clusters", that is, disjoint categories, so that the data can be further used for downstream data analysis tasks. For example, in some photo applications, categories need to be divided according to the content of the user's photos, such as photos of beaches are categorized into one category, photos of mountains are categorized into another category, and so on. However, the user does not manually divide the categories of these photos. At this time, the features are often extracted from these photos, clustered according to the features of the pictures, and then each cluster is defined as a category according to the clustering results, and then the model is trained based on these categories. , for the next step of fine-grained image recognition and other functional applications. At this time, it is often considered: which clustering algorithm is more suitable for a specific scenario? Also, how to measure the performance of these clustering algorithms? In the rest of this paper, we will discuss the following aspects:

- Section 2 describes the performance measures used by clustering algorithms, and explains concepts such as data similarity;
- Section 3 lists the concepts and design ideas of clustering algorithms commonly used in three different scenarios;
- Section 4, based on the basis of traditional clustering algorithms, will show some frontier clustering algorithms ;
- Section 5, the summary of clustering algorithms.

2. PERFORMANCE METRICS

The quality of the clustering algorithm directly affects the quality of downstream data analysis tasks, so how to measure the performance of the clustering algorithm is

crucial. Generally speaking, the core performance index of clustering is to make the similarity within the cluster as high as possible, and make the similarity between the clusters as low as possible. Similarity, also sometimes referred to as "data distance", is a measure of the distance between data. Assuming that x_i, x_j are vectors corresponding to two samples, the following are the commonly used similarity calculation methods in clustering:

- Minkowski distance:

$$dist_{mk}(x_i, x_j) = (\sum_{u=1}^{n} |x_{iu} - x_{ju}|^p)^{\frac{1}{p}}$$

- Euclidean distance:

It is the Euclidean distance when p=2 of the Minkowski distance:

$$dist_{ed}(x_i, x_j) = (\sqrt{\sum_{u=1}^{n} |x_{iu} - x_{ju}|^2})$$

- Manhattan distance:

It is Manhattan distance when p=1 of Minkowski distance:

$$dist_{mk}(x_i, x_j) = (\sum_{u=1}^{n} |x_{iu} - x_{ju}|)$$

- Cosine similarity:

$$cos(\alpha) = \frac{x_i^T x_j}{\|x_i\| \|x_j\|}$$

The result of the calculation is the angle between the two vectors. Therefore, the smaller the value, the higher the similarity between the two vectors. Otherwise, it is an obtuse angle, and the similarity is lower.

- Pearson correlation coefficient:

$$p = \frac{cov(x_i, x_j)}{\sqrt{D(x_i)}\sqrt{D(x_j)}}$$

where $cov(x_i, x_j)$ is the covariance of the two vectors and $D(x_i)$ is the variance of the vectors. The value range of the Pearson correlation coefficient is between [-1, 1]. The closer the value is to 0, the less relevant the surface vector is. The more it tends to -1, the more negatively correlated the vector is, and the more it tends to 1, the more positive the vector is.

- Mahalanobis distance:

$$m = \sqrt{(x_i - x_j)^T S^{-1} (x_i - x_j)}$$

where S is the covariance matrix corresponding to the two vectors.

The above measures are often used for continuous properties, or more precisely called "ordered properties". For discrete and disordered attributes in the sample, for example, the value of the attribute is subject category = {physics, chemistry, law, economics}, how to measure the similarity between them? Consider using VDM (Value Difference Metric) to measure:

$$VDM_p(a, b) = \sum_{i=1}^{k} |\frac{m_{u,a,i}}{m_{u,a}} - \frac{m_{u,b,i}}{m_{u,b}}|^p$$

where $m_{u,a}$ represents the number of samples whose value is a on attribute u, and $m_{u,a,i}$ represents the number of samples whose value is a on attribute u in the ith cluster. At this time, Minkowski distance and VDM can be combined to unify the calculation of sample similarity with ordered and disordered attributes, as follows:：

$$distAll_p(x_i, x_j) = (\sum_{u=i}^{n_c} |x_{iu} - x_{ju}|^p + \sum_{u=n_c+1}^{n} VDM_p(x_{iu}, x_{ju}))^{\frac{1}{p}}$$

with these distance metrics above, performance metrics can be made.

Performance metrics are divided into two categories: external metrics and internal

metrics. An external indicator compares the clustering results with a reference model, such as a reference model defined by human experts. The internal index directly examines the clustering results without using the external reference model.

2.1 External metrics

Due to the intervention of the external reference model, the clustering algorithm may divide each sample into corresponding clusters, which may be inconsistent with the external reference model. First define the confusion matrix of Table 1 to facilitate the explanation of the problem.

Table 1 Definition of Confusion Matrix for binary classification

		Acutal Value	
		Positive	Negative
Predicted Value	Positive	TP	FP
	Negative	FN	TN

Then there are the following external metrics:

- Jaccard index, also known as Jaccard similarity:

$$JC = \frac{TP}{TP + FP + FN}$$

- FM Index (Fowlkes and Mallows Index)：

$$FMI = \sqrt{\frac{TP}{TP + FP} \Delta \frac{TP}{TP + FN}}$$

- Rand Index：

$$RI = \frac{2(TP + NP)}{N(N-1)}$$

where N is the number of samples.

Obviously, the results of the above performance measures are all in the [0,1] interval, the larger the value, the better.

2.2 Internal metrics

Assuming that the clusters of the clustering results are divided into C={C_1, C_2, ..., C_k} , first define the following calculation amount:

- Average distance of samples within cluster C:

$$avg(C) = \frac{2}{|C|(|C|-1)} \sum_{1 \le i \le j \le |C|} dist(x_i, x_j)$$

- The furthest distance between samples in cluster C:

$$diam(C) = \max_{1 \le i \le j \le |C|} dist(x_i, x_j)$$

- The distance between the closest samples of two clusters:

$$d_{min}(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} dist(x_i, x_j)$$

- The distance between the central points of the two clusters:

$$d_{central}(C_i, C_j) = dist(\mu_i, \mu_j), \mu = \frac{1}{|C|} \sum_{1 \le i \le |C|} x_i$$

From this, the following commonly used internal metrics can be derived:

- DB Index (Davies-Bouldin Index):

$$DBI = \frac{1}{k} \sum_{i=1}^{k} \max_{j \ne i} \left(\frac{avg(C_i) + avg(C_j)}{d_{cen}(\mu_i, \mu_j)} \right)$$

Obviously, the smaller the value, the better.

- Dunn Index

$$DI = \min_{i \le i \le k} \left\{ \min_{j \ne i} \left(\frac{d_{min}(C_i, C_j)}{\max_{1 \le l \le k} diam(C_l)} \right) \right\}$$

which is the opposite, the larger the value, the better.

The distance metrics and performance metrics mentioned above are only the more commonly used metrics in clustering. In actual research, these are not the only metrics. For example, F value, mutual information, etc. are

also metrics that cannot be ignored. Like Xing et al [1] mentioned a distance metric learning method, which is embedding into the clustering process for learning. We will not be introduced more here.

3. CLUSTERING ALGORITHM

There are many types of clustering algorithms. The following will review the three main types of commonly used clustering algorithms according to the organization of data clustering.

3.1 Based on partition

This type of clustering algorithm generally adopts a greedy strategy. By defining an optimal objective function, the sample points are continuously classified into the categories of the K representative points through an iterative method, and the clustering effect is gradually improved.

K-means[2] is a typical representative of them. Its objective function is to minimize the squared error of all sample points:

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|_2^2$$

The algorithm first needs to specify the number of clusters K to be divided, randomly select K representative points as the initial vector, and then iteratively determine the distance between each sample point and these representative points, and assign the category to which the representative point with the smallest distance belongs. Give the sample points in the iteration, and each iteration, recalculate the representative point as the center point of the sample point to which it belongs as a new representative point, and iterate until all the sample points are not updated. Therefore, as can be seen from Fig.1, K-means is a process of continuously gathering samples at the best cluster center point (representative point). The advantages of K-means are: the Euclidean distance is easy to calculate, the algorithm does not depend on the order of the data, and so on. Its shortcomings are also quite obvious, mainly including:

- The number of clusters should be pre-specified, but it is difficult to specify the appropriate number of clusters K;
- Sensitive to outliers;
- It is less effective for linearly inseparable data. These are also often inherent shortcomings of partition-based clustering algorithms.

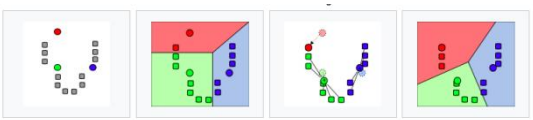

Fig.1. Procedure of K-means, from left to right. (picture by wikipedia).

K-medoids is an improvement of the K-means algorithm. The biggest difference between it and K-means is that the method of selecting the center point: the center point selected by K-medoids must be a sample point each time, while the center point selected by K-means Can be a point other than the sample point. When K-medoids selects the center point, it needs to traverse all the points in the cluster, and select the point with the smallest sum of distances from all other points to the center point in the previous cluster as the center point. In this way, K-medoids can greatly reduce the impact of outliers on clustering results.

There is also a commonly used method based on probability statistics, which also belongs to the clustering algorithm based on partition. It can be assumed that there is a hidden category distribution in the data space, the number of categories in the distribution is K, and the data samples are generated through these K categories, that is, the K clusters: first select a cluster with a certain probability, and

then select a sample with a certain probability in that cluster. So the next step is to calculate the likelihood of these K clusters and their probabilistically generated dataset. It is assumed that the samples are sampled independently, so for the dataset D={D_1, D_2, D_3, …} we have:

$$P(D|C) = \prod_{i=1}^{n} P(D_i|C) = \prod_{i=1}^{n} \sum_{j=1}^{k} \alpha_j P(D_i|\theta)$$

where α_j is the probability of selecting each cluster, so the total probability of selecting K clusters is 1, that is $\sum_{j=1}^{k} \alpha_j = 1$, and θ is the parameter of the specified probability distribution. For example, in practical applications, Gaussian distribution can be selected, then the parameter is the n-dimensional mean vector and the n x n order coordinator variance matrix. Parameter estimation can use the EM algorithm [3] or other methods. After the parameters are obtained, the cluster class which the sample belongs to can be obtained through the posterior probability $P(C_i \mid x_i)$.

Table 2 summarizes the time complexity of the partition-based clustering algorithm, where GMM [4] is a Gaussian mixture distribution clustering algorithm, k is the number of clusters, t is the number of iterations, and n is the number of samples.

Table.2. Time complexity for partition alorithm

K-means	K-mediods	GMM
O(k*t*n)	$O(k(n-k)^2)$	$O(k*t*n^2)$

3.2 Based on density

This type of algorithm focuses on the degree of closeness between samples, such as connectivity, and based on connectivity, the samples are continuously expanded into clusters to obtain the final clustering effect. Among them, DBSCAN[5] is a typical representative of this clustering algorithm. DBSCAN requires predetermined parameters (ϵ, MinPts), and the following concepts are defined:

- ϵ-neighborhood, that is $N_\epsilon(x_j) = \{x_i \in D | dist(x_i, x_j) \leq \epsilon\}$;
- core object: If x_j's ϵ -neighborhood contains at least MinPts samples, then x_j is a core object;
- directly density-reachable: If x_j is located in x_i's-neighborhood, and x_i is a core object, then It is said that x_j is directly reached by the density of x_i;
- density-reachable: For x_i and x_j, if there is a sample sequence p_1, p_2,...,p_n, where $p_1=x_i$, $p_n=x_j$, and p_{i+1} is directly density-reachable by p_i , then it is said that x_j is density-reachable by x_i;
- density-connected: For x_i and x_j, if there is x_k such that both x_i and x_j are density-reachable by x_k, then x_i and x_j are said to be density-connected.

Based on the above definition, specifically, the algorithm finds all core objects according to the given parameters (ϵ, MinPts) takes any core object as the starting point, finds out the density-reachable samples to generate clusters, and iterates continuously to all the core objects. Core objects have been accessed so far. As shown in Fig.2, the algorithm can be regarded as a process of continuously expanding the connected area from seed points (ie, core objects) to form clusters. The main advantage of DBSCAN is that it can be used for linearly inseparable data sets, and its clustering effect is generally better, and it does not need to indicate the number of clusters to be clustered. However, it needs to specify two parameters (ϵ, MinPts) that are difficult to apply to the sample data set, and even different parts of the data set may need to specify different parameters to achieve a good clustering effect, so it is too sensitive to parameters.

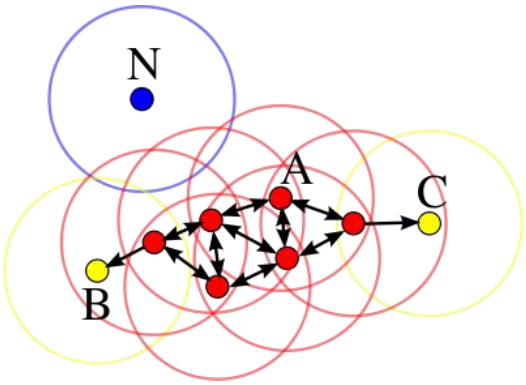

Fig.2. Procedure of DBSCAN.(picture by wikipedia)

OPTICS[6] is an improved algorithm of DBSCAN, which also needs to specify the same two parameters (ϵ, MinPts), but is no longer sensitive, as long as MinPts is determined, slight changing in ϵ do not affect the clustering results. In addition to inheriting the definition of DBSCAN, OPTICS has two more definitions：

- core distance: The minimum radius that makes a sample point a core point (ie, core object). Under the premise of a given neighborhood radius and MinPts parameters, the core distance can be smaller than the given ϵ value;
- achievable distance: sample point distance to the core point.

The algorithm process is as follows:

(1) Create two queues: the pending queue is used to store the core points and their direct density points, and they are arranged in ascending order of reachable distance; the result queue is used to store the output order of the sample points. The points in the result queue are the processed samples;

(2) Select an unprocessed core point, put it into the result queue, calculate the reachable distance of the sample points in the neighborhood, and place the sample points in ascending order of reachable distance. into the pending queue.

(3) If the queue to be processed is empty, go back to step (2) to reselect the processing data. Otherwise, extract the first sample from the pending queue, if it is a core point, calculate the reachable distance, and put the point with the smallest reachable distance into the result queue. If it is not the core point, repeat step (2); if there is already a direct density reachable point in the queue to be processed, if the new reachable distance is smaller than the old reachable distance at this time, the old reachable distance is replaced by the new reachable distance , the pending queue is reordered (because one object may be reachable by multiple core objects); if the direct density-reachable sample point does not exist in the pending queue, insert this point and reorder the ordered queue;

(4) iterates (2), (3) until all points in the data set are processed, then the algorithm ends, and the ordered sample points in the result queue are output.

OPTICS does not explicitly generate clusters, but rather generates an ordering of clusters that represents the density-based clustering structure of each sample point.

The time complexity of DBSCAN under violent iteration is $O(n^2)$. Its author mentioned in the paper [5] that R*-tree [7] can be used as a data structure, which is a balanced tree in a multi-dimensional space. So time complexity DBSCAN and OPTICS can both be optimized to O(n*logn).

3.3 Based on hierarchy

The hierarchical clustering algorithm divides the data set at different levels, using a bottom-up aggregation strategy or a top-down splitting strategy to make the data set form a tree structure. Example of hierarchical clustering is show as Fig.3.

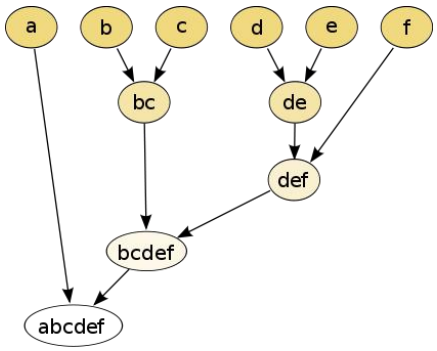

Fig.3. Example of hierarchical clustering for strings (picture by wikipedia).

AGNES[8] is a classic bottom-up hierarchical clustering algorithm. It first specifies the number of clusters to be clustered, then treats each sample in the data set as an initial cluster, and then finds the two closest clusters at each step of the algorithm to merge them. Iteration is repeated continuously until the preset number of clusters is reached. For the calculation of the distance between clusters, please refer to Section 2. Generally, the minimum distance (determined by the closest sample of the two clusters), the maximum distance (determined by the farthest sample of the two clusters), and the average distance can be measured (determined by the two clusters cluster all samples determined). The advantages of hierarchical clustering are generally as follow:

- It is highly analyzable, forming a tree-shaped cluster analysis graph to understand the hierarchy of clustering;
- It can be applied to any similarity and distance measures;
- It has strong applicability to any data type and is easy to expand.

Its disadvantages are generally such as:

- It is difficult to choose an appropriate time to stop iteration, because it is also necessary to manually specify the number of clusters for clustering;
- Once the clusters are merged, they cannot be undone;
- The time complexity is high.

The CURE algorithm [9] is an improvement to AGNES. The difference from AGNES is that it does not use all points or center points plus distance to represent a cluster, but extracts a fixed number of points from each cluster with better distribution. as a representative point for this cluster. The algorithm process is as follows:

(1) At the beginning, each sample point is also called a cluster;

(2) From each cluster, select a small part as the representative points of the cluster, and the selected points are as far apart as possible;

(3) Move each representative point a distance: the distance from its position to the center of the cluster is multiplied by a shrinkage factor, such as 0.2, to make them closer to the center point; the calculation formula of the center point of the cluster and the representative point is as follows:

$$w.mean = \frac{|u| \cdot u.mean + |v| \cdot v.mean}{|u| + |v|}$$

$$w.rep = p + \alpha(w.mean - p)$$

where w is the merged cluster, u and v are the two clusters to be merged, is the shrinkage factor, and p is the sample point to be investigated;

(4) The distance between clusters is defined as the difference between any two representative points after the move;

(5) When the distance between a pair of representative points of the two clusters is less than the specified threshold, the two clusters are merged, and the process is repeated until there are no close enough clusters;

(6) Traverse each Sample points, compare them with representative points, and assign points to the nearest cluster.

Due to the use of representative points, the

model adds a certain randomness, which can reduce the influence of noise data on clustering, and its shrinkage characteristics can adjust model matching The application scenario of non-Gaussian distribution, but the disadvantage is still high time complexity, which is $O(n^2)$ like AGNES.

4. FRONTIER RESEARCH

With the development of big data, the amount of data is getting larger and larger, and the sample space is also higher-dimensional, which requires the design of clustering algorithm to complete the processing of data while ensuring a certain time complexity. In recent years, some novel clustering algorithms have attracted more and more attention.

Kernel clustering algorithm uses nonlinear mapping to map sample data from input space to a higher-dimensional feature space for cluster analysis. The advantage of mapping data to high-dimensional space is that the data can become linearly separable from the original linear inseparability. Therefore, the combination of traditional clustering algorithm and kernel method may make the algorithm more efficient and more analytic. The kernel SOM algorithm proposed by MacDonald D et al. [10] is a collection of the kernel method and the traditional SOM [11] clustering method. Also, Zhong-dong Wu et al. [12] proposed that kernel FCM is also a kernel clustering method that combines the kernel method with FCM [13]. Another example is the MKC method proposed by Bin Zhao et al. [14], which is an improvement of the MMC [15] method. MMC tries to find a maximally spaced hyperplane for clustering, while MKC can simultaneously find the maximally spaced hyperplane, the optimal clustering label and the optimal kernel.

The ensemble clustering algorithm draws on the ensemble learning in machine learning. In order to improve the accuracy and stability of the clustering results, a better result can be produced by integrating the results of multiple base clustering algorithms. A clustering algorithm model can produce good results for a specific data set, but may have poor effect on other data sets. In this case, multiple independent base clusterers are needed to cluster the original data set, and then use An ensemble method improves the clustering results and finally obtains a better clustering effect. The process of the ensemble clustering algorithm is shown in Fig.4. It can be seen that the key to ensemble clustering is to find a suitable Consensus function, that is, a combination strategy to map multiple clustering results to one. There are various schemes for the selection of consensus functions, such as based on correlation matrix, based on graph partitioning, based on relabeling and voting, based on genetic algorithm, and so on. The VM model proposed by Evgenia Dimitriadou, et al. [16] is an ensemble clustering method based on voting. Each data point has a degree of belonging to the cluster, and finally the clustering result with the largest degree of membership is selected as the final result. The EAC method proposed by ALN Fred et al. [17] forms a proximity matrix from the results of different cluster divisions. The elements of the matrix measure the average number of samples that belong to the same cluster in different cluster divisions. This matrix performs hierarchical clustering algorithm to get the final clustering result.

Dong Huang et al. [18] proposed a clustering method that can utilize diversified metrics to combine multiple random subspaces, called MDEC, which generates a large number of diversified metrics by randomizing a scaled exponential similarity kernel. These metrics are then coupled with the random subspaces to form a large set of metric-subspace pairs,

and these sets are then clustered using a spectral clustering algorithm.

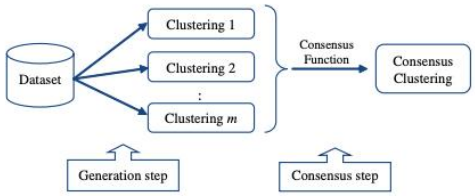

Fig.4. general process of ensemble clustering.[19]

In addition, the combination of current clustering algorithms and other disciplines has also produced many new algorithms in specific fields, such as clustering based on quantum theory, clustering based on graphs, clustering based on swarm intelligence, etc. They all are prospective.

5. CONCLUSION

Clustering is a field with many algorithms and the fastest development in machine learning. An important reason is that there is no objective classification standard due to the lack of data class identification. For a specific application scenario, given a data set, a new clustering algorithm can always be designed by using different representations of the tightness of the sample distribution. The main purpose of this paper is to expound the basic and core design ideas of various classical clustering algorithms, and to let readers understand the importance and value of clustering in data mining by comparing the advantages and disadvantages of various algorithms, which is very important for data mining. A systematic and general study of the theory is very helpful.

REFERENCES

[1] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," p. 8.

[2] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979, doi: 10.2307/2346830.

[3] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. John Wiley & Sons, 2007.

[4] C. Rasmussen, "The Infinite Gaussian Mixture Model," in *Advances in Neural Information Processing Systems*, 1999, vol. 12. Accessed: Jul. 08, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/1999/hash/97d98119037c5b8a9663cb21fb8ebf47-Abstract.html

[5] M. Ester, H.-P. Kriegel, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," p. 6.

[6] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure," p. 12.

[7] N. Beckmann, R. Schneider, B. Seeger, and P. Informatlk, "The R-Tree: An Efficient and Robust Access Method for Points and Rectangles," p. 10.

[8] L. Kaufman and P. J. Rousseeuw, Eds., "Agglomerative Nesting (Program AGNES)," in *Wiley Series in Probability and Statistics*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 1990, pp. 199–252. doi: 10.1002/9780470316801.ch5.

[9] S. Guha, R. Rastogi, and K. Shim, "CURE: an efficient clustering algorithm for large databases," *SIGMOD Rec.*, vol. 27, no. 2, pp. 73–84, Jun. 1998, doi: 10.1145/276305.276312.

[10] D. MacDonald and C. Fyfe, "The kernel self-organising map," in *KES'2000. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. Proceedings (Cat. No.00TH8516)*, 2000, vol. 1, pp. 317–320 vol.1. doi: 10.1109/KES.2000.885820.

[11] T. Kohonen, *Self-Organizing Maps*. Springer Science & Business Media, 2012.

[12] Z. Wu, W. Xie, and J. Yu, "Fuzzy C-means

clustering algorithm based on kernel method," in *Proceedings Fifth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2003*, Sep. 2003, pp. 49–54. doi: 10.1109/ICCIMA.2003.1238099.

[13] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2–3, pp. 191–203, Jan. 1984, doi: 10.1016/0098-3004(84)90020-7.

[14] B. Zhao, J. T. Kwok, and C. Zhang, "Multiple Kernel Clustering," in *Proceedings of the 2009 SIAM International Conference on Data Mining*, Apr. 2009, pp. 638–649. doi: 10.1137/1.9781611972795.55.

[15] E. Stone and J. Ayroles, "Modulated modularity clustering as an exploratory tool for functional genomic inference.," *PLoS Genet*, vol. 5, no. 5, p. e1000479, 2009.

[16] E. Dimitriadou, A. Weingessel, and K. Hornik, "Voting-Merging: An Ensemble Method for Clustering," in *Artificial Neural Networks — ICANN 2001*, vol. 2130, G. Dorffner, H. Bischof, and K. Hornik, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 217–224. doi: 10.1007/3-540-44668-0_31.

[17] 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, doi: 10.1109/TPAMI.2005.113.

[18] D. Huang, C.-D. Wang, J.-H. Lai, and C.-K. Kwoh, "Toward Multidiversified Ensemble Clustering of High-Dimensional Data: From Subspaces to Metrics and Beyond," *IEEE Transactions on Cybernetics*, pp. 1–14, 2021, doi: 10.1109/TCYB.2021.3049633.

[19] S. Vega-Pons and J. Ruiz-Shulcloper, "A SURVEY OF CLUSTERING ENSEMBLE ALGORITHMS," *Int. J. Patt. Recogn. Artif. Intell.*, vol. 25, no. 03, pp. 337–372, May 2011, doi: 10.1142/S0218001411008683.