Article

Divergence-Based Locally Weighted Ensemble Clustering with Dictionary Learning and $L_{2,1}$-Norm

Jiaxuan Xu 1, Jiang Wu 1, Taiyong Li 1,∗ and Yang Nan 2

1 School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu 611130, China
2 Department of Computer Science, Harbin Finance University, Harbin 150030, China
∗ Correspondence: litaiyong@gmail.com

Abstract: Accurate clustering is a challenging task with unlabeled data. Ensemble clustering aims to combine sets of base clusterings to obtain a better and more stable clustering and has shown its ability to improve clustering accuracy. Dense representation ensemble clustering (DREC) and entropy-based locally weighted ensemble clustering (ELWEC) are two typical methods for ensemble clustering. However, DREC treats each microcluster equally and hence, ignores the differences between each microcluster, while ELWEC conducts clustering on clusters rather than microclusters and ignores the sample–cluster relationship. To address these issues, a divergence-based locally weighted ensemble clustering with dictionary learning (DLWECDL) is proposed in this paper. Specifically, the DLWECDL consists of four phases. First, the clusters from the base clustering are used to generate microclusters. Second, a Kullback–Leibler divergence-based ensemble-driven cluster index is used to measure the weight of each microcluster. With these weights, an ensemble clustering algorithm with dictionary learning and the $L_{2,1}$-norm is employed in the third phase. Meanwhile, the objective function is resolved by optimizing four subproblems and a similarity matrix is learned. Finally, a normalized cut (Ncut) is used to partition the similarity matrix and the ensemble clustering results are obtained. In this study, the proposed DLWECDL was validated on 20 widely used datasets and compared to some other state-of-the-art ensemble clustering methods. The experimental results demonstrated that the proposed DLWECDL is a very promising method for ensemble clustering.

Keywords: clustering; ensemble clustering; $L_{2,1}$-norm; similarity; subspace clustering; dictionary learning

1. Introduction

For a long time, clustering has been widely studied as an important technology for machine learning [1–4]. However, due to the lack of prior knowledge, i.e., pre-label training, the accuracy of clustering algorithms is much lower than that of supervised learning methods. Traditional single clustering methods, such as k-means, balanced iterative reducing and clustering using hierarchies (BIRCH), density-based spatial clustering of applications with noise (DBSCAN), etc., cannot usually achieve good clustering results for complex data [5,6]. Encouraged by the accuracy improvement effects of ensemble learning methods, many researchers have begun to study clustering ensemble algorithms. Clustering ensembles learn from multiple base clustering results to obtain consensus results, which can greatly improve the clustering accuracy without the need for prior knowledge [7–12].

The focuses of ensemble clustering methods are either the selection of base clustering or ensemble methods [13]. The selection of base clustering has two influences on the consensus results: accuracy and diversity. Higher accuracy usually leads to the lower diversity of the base clustering, while higher diversity results in the lower accuracy of the base clustering [14]. Therefore, balancing these two factors is key in the selection of base clustering. Ensemble methods aim to learn more robust consensus results by mining...
more effective information from the base clustering sets. Essentially, ensemble methods mine more inner information from the base clusterings. Although there are many robust ensemble methods, it is difficult to identify which ensemble method outperforms the others on a given dataset due to the randomness of the base clustering selection and the diversity of datasets.

Generally speaking, the most commonly used representative methods for mining this information from base clusterings include (1) co-association (CA) matrices, which represent the mutual relationships between samples in the base clustering sets, i.e., relationships at the sample level, (2) cluster–cluster (CC) matrices, which indicate the relationships between clusters in base clustering sets, i.e., relationships at the cluster level, and (3) sample–cluster matrices, which represent the relationships between samples and clusters in base clustering sets, i.e., relationships at the sample–cluster level. Both CA and CC matrices can be calculated using sample–cluster matrices. CA matrices reveal the probability that samples are of the same class. The larger the value of $X_{ij}$ in a CA matrix, the greater the possibility that the samples $i$ and $j$ are of the same class. Some methods aim to retain or learn reliable samples in CA matrices and then seek consensus results [10,14]. For example, Jia et al. proposed an effective self-enhancement framework for CA matrices to improve the ensemble clustering results, through which high-confidence information was extracted from base clusterings [15]. CC matrices reveal the similarities between clusters, which cannot be used for ensembles alone due to the lack of effective information, so it has to be combined with other valid information to perform accurate clustering. Therefore, some researchers have used CC matrices to calculate similarities and then mapped them as weights to CA matrices or sample–cluster matrices [11,16]. Sample–cluster matrices are the original matrices in base clustering sets and retain the most complete information in base clustering sets. Some methods choose to explore hidden information in the original matrices [11]. For example, based on sample–cluster matrices, the dense representation ensemble clustering (DREC) method introduces microcluster representation, reduces the amount of data, retains the effective information from sample–cluster matrices to the greatest extent and then performs dense representation clustering, which not only improves the time performance but also explores the hidden effective information to the greatest extent [13]. Huang et al. pointed out that the differences between microclusters also play important roles in ensemble clustering [17]. However, the DREC method ignores the differences between microclusters. Moreover, it does not reveal the underlying structures in sample–cluster matrices well. Entropy-based locally weighted ensemble clustering (ELWEC) has been demonstrated as being effective in improving clustering accuracy [18]. The key reason for this is the adoption of the idea of mapping entropy-based local weights to clustering. However, the ELWEC method measures the weights of clusters rather than microclusters and ignores sample–cluster relationships, thereby limiting the clustering performance to some extent. Very recently, the Markov process [19], a growing tree model [20], a low-rank tensor approximation [21] and an equivalence granularity [22] have been applied to ensemble clustering to achieve better clustering results.

Motivated by the above analysis, a divergence-based locally weighted ensemble clustering with dictionary learning (DLWEC) is proposed in this paper. The idea of local weights was introduced to the DLWEC. Different from the entropy-based local weights of clusters in ELWEC, this study used the divergence-based local weights of microclusters for ensemble clustering. Specifically, low-rank representation, the $L_{2,1}$-norm and dictionary learning were applied to design the objective function and the corresponding constraints. We used the augmented Lagrange multiplier (ALM) with alternating direction minimization (ADM) strategy for the optimization of the objective function. Extensive experiments on real datasets demonstrated the effectiveness of our proposed method.

The main contributions of this paper are summarized as follows:

1. The proposal of a Kullback–Leibler divergence-based weighted method to better reveal relationships between clusters;
(2) The use of low-rank representation instead of dense representation to better explore hidden effective information and low-rank structures of original matrices;
(3) The application of the $L_{2,1}$-norm to noise to improve robustness;
(4) The introduction of adaptive dictionary learning to better learn low-rank structures;
(5) Extensive experiments to demonstrate that the proposed DLWECSDL can significantly outperform other state-of-the-art approaches.

The rest of this paper is organized as follows. Section 2 reviews related works on ensemble clustering. The proposed ensemble clustering method is described in detail in Section 3. The experimental settings and results are analyzed and discussed in Section 4. Finally, Section 5 concludes the paper and provides our recommendations for future work.

2. Related Works

2.1. Ensemble Clustering

The goal of ensemble clustering is to find consensus results based on $M$ base clusterings. To obtain good consensus results, two questions naturally arise. The first question is the selection of the base clusterings, which should not only ensure the diversity of the base clusterings but also the quality or accuracy of the base clusterings. Existing studies have proposed some methods that take into account the diversity and quality of base clusterings [23,24]. The second question is the ensemble method, which is roughly divided into two categories: similarity matrix-based learning and graph-based learning. Similarity matrices are the core problems in various clustering methods. In ensemble clustering, similarity matrices are obtained by exploring sample–sample, cluster–cluster and sample–cluster relationship matrices and then using spectral clustering to obtain the final clustering results.

Based on similarity matrices, our method follows a dense representation ensemble clustering framework, finds microclusters and then performs dense representation at the microcluster level. However, it does not work for microclusters that contain more samples. Therefore, we designed a local weight-based microcluster ensemble method and used a new low-rank representation clustering method. Inspired by the ALRR method [25], we introduced the $L_{2,1}$-norm and adaptive dictionary learning to the new low-rank representation method.

2.2. Microcluster Representatives

Our approach starts by finding microcluster representatives to simplify the problem. A sample–cluster matrix needs to be reconstructed before looking for these microcluster representatives.

Figure 1 is an example that illustrates our definition of a microcluster, where $C_i$ is the $i$-th base clustering, $X_j$ represents the $j$-th sample and the numbers 1–7 in the heading of the full data matrix are the global renamed cluster IDs. We reconstructed the original base clustering results to obtain the full data matrix, in which we observed that the information in samples $X_1$ and $X_2$ was completely consistent. Therefore, we grouped $X_1$ and $X_2$ into the same microcluster and chose either $X_1$ or $X_2$ as the microcluster representative.
2.3. Information Entropy-Based Locally Weighted Method

The information entropy-based locally weighted method mainly explores the uncertainty of each cluster [18]. It introduces the concept of entropy to calculate the uncertainty of each cluster and then determines the weight of each cluster using a monotonically decreasing function. It forms results based on the more stable cluster, the smaller the uncertainty and the larger the weight. However, for similar clusters, it cannot guarantee that the final weights are consistent, even though the weights of completely different clusters may be consistent.

We used the locally weighted method for microclusters to calculate the weight of each cluster in each base clustering and then apply it to the microclusters. The weights were measured using the ensemble-driven cluster index (ECI).

Taking the cluster in the $i$-th base clustering $\pi_i$ as an example, the weights were calculated as follows:

$$H(\pi_i) = \sum_{m=1}^{M} - \sum_{j=1}^{K} p(\pi_i, \pi_j^m) \log_2 p(\pi_i, \pi_j^m)$$  \hspace{1cm} (1)

$$p(\pi_i, \pi_j^m) = \frac{|\pi_i \cap \pi_j^m|}{|\pi_i|}$$  \hspace{1cm} (2)

$$ECI(\pi_i) = e^{-\frac{H(\pi_i)}{\theta}}$$  \hspace{1cm} (3)

where $-$ represents the subtraction operation, $\theta$ is a control parameter and $|C_i|$ represents the number of samples in $C_i$. After obtaining the ECI weight of each cluster, we applied them to the selected representative microcluster matrix to obtain the final data matrix.

2.4. Dense Representation Ensemble Clustering

The concept of microclusters has been introduced into the DREC method. The scale of ensemble clustering problems is simplified using the “slim-down strategy” and then similarity matrices can be obtained using the dense representation method and the final result segmentation can be obtained by applying the Ncut algorithm. Because of the microclusters, the DREC method improves time efficiency and preserves more original information. However, the DREC method treats “shrunk” samples equally, which does not work for microcluster samples. At the same time, although the DREC method considers the influence of noise, it fails to consider the selection of the base clusterings, which also leads to the instability of the final results of the randomly selected base clustering integration.

3. Divergence-Based Locally Weighted Ensemble Clustering with Dictionary Learning (DLWECDL)

The goal of ensemble clustering is to learn consistent results based on $M$ base clusterings. In ensemble clustering, the key is to explore the effective information in base
clustering sets. The effective information in base clustering sets is hidden within three common manifestations, namely sample–sample relational representation, sample–cluster relational representation and cluster–cluster relational representation. We believe that good consensus results can be obtained when all of the valid information from the three representations can be fully utilized. Sample–cluster relationship matrices are key to linking these three representations because they can be used to calculate the remaining two representations. Therefore, we took the sample–cluster relational representation as the base representation and used it as the data matrix for our method. It was the original representation of our base clustering set.

3.1. Divergence-Based Locally Weighted Method

The information entropy-based locally weighted method mainly considers the uncertainty between clusters. We introduced the Kullback–Leibler (KL) divergence, which is widely used to measure the differences between distributions. When distributions are exactly the same, the KL divergence is 0. Considering the good performance of KL divergence in some clustering methods over recent years, we introduced KL divergence as a measure of local weights. Since $p(\pi_i, \pi_m^j)$ and $p(\pi_m^j, \pi_i)$ are not clear probabilistic interpretations, the KL divergence results here were not guaranteed to always be greater than 0. After obtaining the KL divergence, we used the ECI entropy mapping function to obtain the new KL divergence weights.

$$KL(\pi_i) = \sum_{m=1}^{M} \sum_{j=1}^{K} p(\pi_i, \pi_m^j) \log_2 \frac{p(\pi_i, \pi_m^j)}{p(\pi_m^j, \pi_i)}$$ (4)

To better illustrate the advantages of KL divergence weighting, an example is presented in Figure 2, where $C_i$ represents the $i$-th base clustering result, $\pi_i^j$ denotes the $j$-th cluster in the $i$-th base clustering and the numbers 1–12 in the circles are the numbers of the samples. As shown in Table 1, we compared the results of the inter-cluster entropy calculation and the KL divergence calculation, where $R$ represents the ratio of the maximum number of samples in the stable subsets to the number of samples that were contained in the clusters. For example, Samples 1, 2 and 3 were assigned to $\pi_1^1$, $\pi_2^1$ and $\pi_3^1$ in the base clusterings $C_1$, $C_2$ and $C_3$, respectively. The three samples were classified into the same class in different base clustering results. This meant that the most stable subsets of $\pi_1^1$, $\pi_2^1$ and $\pi_3^1$ were Sample 1, Sample 2 and Sample 3, respectively. Therefore, the R values for clusters $\pi_1^1$, $\pi_2^1$ and $\pi_3^1$ were $\frac{1}{3} = 1$, $\frac{2}{3} = 0.6$ and $\frac{3}{3} = 0.75$, respectively. It can be observed from Table 1 that the R values of $\pi_1^4$, $\pi_3^4$ and $\pi_2^3$ were consistent but the entropy values were quite different. This led to inconsistent weights. The same situation occurred in $\pi_2^2$ and $\pi_2^3$. The KL divergence method reduced the gaps between clusters with the same R values as much as possible so that the weights were as consistent as possible.
Figure 2. An example of the entropy and divergence value calculations.

Table 1. The entropy values versus the KL divergence values.

| Cluster | \( \pi_1 \) | \( \pi_2 \) | \( \pi_3 \) | \( \pi_4 \) | \( \pi_5 \) | \( \pi_6 \) | \( \pi_7 \) | \( \pi_8 \) | \( \pi_9 \) | \( \pi_{10} \) |
|---------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| R       | 1           | 0.33        | 0.67        | 0.67        | 0.6         | 0.5         | 0.67        | 0.75        | 0.5         | 0.25        |
| Entropy | 0           | 1.837       | 1.837       | 0.918       | 2.342       | 2.500       | 0.918       | 1.623       | 3.000       | 2.000       |
| Divergence | 0       | −0.650      | −0.650      | −0.288      | 0.734       | 0.288       | −0.288      | 0.120       | 0.248       | 0.432       |

3.2. \( L_{2,1} \)-Norm Subspace Clustering of Adaptive Dictionaries

After obtaining the final data matrix, we developed a new subspace clustering method. Unlike dense representation, we explored similarity matrices using low-rank representation, which incorporated an adaptive dictionary learning strategy and employed a new regularization term, i.e., the \( L_{2,1} \)-norm.

The original low-rank subspace clustering that could explore similarity matrices was formulated as follows:

\[
\min_{Z,E} \|Z\|_* + \lambda \|E\|_{2,1}
\]

\[
\text{s.t. } X = DZ + E
\]

(5)

where \( \lambda \) is a regularization parameter, \( X \) represents the data matrix, \( D \) is the dictionary, \( Z \) is the low-rank representation coefficient matrix, \( E \) is the noise and \( \| \cdot \|_* \) and \( \| \cdot \|_{2,1} \) represent the nuclear norm and the \( L_{2,1} \)-norm of the matrix, respectively. The original low-rank representation method the data \( X \) as a dictionary \( D \). On this basis, many low-rank representation subspace clustering algorithms have been further proposed and the adaptive dictionary learning low-rank representation \([25]\) problem can be formulated as follows:

\[
\min_{Z,E} \|Z\|_* + \lambda \|X - DZ\|_F^2
\]

\[
\text{s.t. } DD^T = I_d
\]

(6)

where \( \| \cdot \|_F \) is the famous Frobenius norm, which was used here for computational convenience because many closed-form solutions that are based on this norm can greatly improve time efficiency. In order to eliminate the arbitrary scaling factor in the process of dictionary learning, \( D \) and \( X \) were replaced by \( P^T X \). To take into account the advantages of dictionary learning and noise immunity, our method was formulated as follows:

\[
\min_{Z,P,E} \|Z\|_* + \lambda \|E\|_{2,1}
\]

\[
\text{s.t. } X = P^T XZ + E \quad P^T XX^T P = I_d
\]

(7)
where \( P \) denotes a low-dimensional projection matrix and \( I_d \) is the identity matrix. The proposed method not only retains dictionary learning in low-rank representation, i.e., learning better and more orthogonal dictionaries, but also adopts the \( L_{2,1} \)-norm to make it more robust to noise. A widely accepted theory is that high-dimensional data are determined by low-dimensional structures. The low-rank matrix \( Z \) that was obtained according to the objective function contained the angle information between the data samples. We performed SVD decomposition on the low-rank matrix \( Z \) and obtained \( H = U \Sigma V^T \). We then used \( H \) to obtain the final similarity matrix \( W \).

The detailed steps of the proposed DLWECDL are described in Algorithm 1, a flowchart for which is also shown in Figure 3. It should be noted that \( a \) in Algorithm 1 is a positive integer parameter and \( h_i \) and \( h_j \) are the \( i \)-th and \( j \)-th rows of matrix \( H \), respectively.

---

**Algorithm 1**: Divergence-based locally weighted ensemble clustering with dictionary learning (DLWECDL).

**Input**: \( M \) base clustering, \([C_1, C_2, \cdots, C_m]\).

**Output**: Consensus clustering result \( S \)

1. Reconstruct the data matrix to obtain a microcluster representative matrix.
2. Calculate the local divergence weight and local entropy weight and weigh the microcluster representative matrix.
3. Learn low-rank structures \( Z \) by Low-rank representation with adaptive dictionary learning and the \( L_{2,1} \)-norm.
4. Calculate \( H \) by SVD decomposition of \( Z \). Calculate similarity matrix \( W \) by \( H \).
   \[
   H = U \Sigma V^T, \quad Z = \text{SVD}(U \Sigma V^T), \quad [W]_{ij} = \left( \frac{h_i h_j^T}{\|h_i\|_2 \|h_j\|_2} \right)^{2a}
   \]
5. Perform Ncut to partition the similarity matrix \( W \).
6. Obtain consensus result \( S \) by microcluster representative label mapping.

---

**Figure 3**: A flowchart of the proposed DLWECDL.
As shown in Figure 3, DLWECDL first introduces microclusters to reduce the amount of data, which reduces data redundancy and improves time efficiency. Then, DLWECDL performs local weighting on the simplified dataset. Two weighting methods, namely entropy-based and KL divergence-based weighting, are used to better represent the micro-clusters. Theoretically, the entropy-based weighting method focuses more on the uncertainty of the clusters themselves while the KL divergence-based method focuses more on the relative uncertainty, i.e., the differences between clusters. This also means that datasets with more diverse base clusterings may be more suitable for the KL divergence-based weighting method. The third step uses low-rank representation with dictionary learning and the $L_{2,1}$-norm to explore deep structures. After using the Ncut method to partition the data, the labels of the reduced dataset need to be mapped to the full dataset because of the introduction of the microclusters.

To demonstrate the feasibility and effectiveness of the proposed algorithm more intuitively, an example on a 2D synthetic dataset is presented in Figure 4. In the example, k-means clustering algorithms with different $k$s were performed 20 times. Their outputs were used to generate the microclusters, from which a matrix of the KL divergence weights was obtained. Then, low-rank representation with adaptive dictionary learning and the $L_{2,1}$-norm was applied to the weighted matrix to obtain an affinity matrix and the corresponding labels for the microclusters. Finally, the labels were mapped to obtain the final results of the proposed DLWECDL. In Figure 4, the microclusters, KL divergence weights, affinity matrix and labels are the intermediate data of the proposed DLWECDL.

![Figure 4. An example on a synthetic dataset.](image)

### 3.3. Optimization Method

For Problem (7), we employed the augmented Lagrange multiplier (ALM) with alternating direction minimization (ADM) strategy for optimization [26]. The auxiliary variable $J$ needed to be introduced here. The augmented Lagrangian function is as follows:

$$L = \|J\|_* + \lambda \|E\|_{2,1} + \text{tr} \left( Y_1^T \left( X - P^T X Z - E \right) \right) + \text{tr} \left( Y_2^T \left( Z - J \right) \right) +$$

$$\frac{\mu}{2} \left( \|X - P^T X Z - E\|_F^2 + \|Z - J\|_F^2 \right),$$

where $Y_1$ and $Y_2$ are Lagrange multipliers and $\mu$ is a penalty parameter. According to the ADM strategy [26], we divided the objective into several subproblems that could be efficiently optimized.
3.3.1. Subproblem $J$

To update $J$, we needed to solve the following problem:

$$J^* = \arg\min_J \frac{1}{\mu} \|J\|_* + \frac{1}{2} \|J - (Z + Y_2 / \mu)\|_F^2$$ (9)

Problem (9) had a popular closed-form solution, which was solved using SVD decomposition. It was consistent with the first solution of the LRR method.

3.3.2. Subproblem $Z$

To update $Z$, we needed to solve the following problem:

$$Z^* = \arg\min_Z \text{tr} \left( Y_1^T (X - P^T XZ - E) \right) + \text{tr} \left( Y_2^T (Z - J) \right) + \frac{\mu}{2} (\|X - P^T XZ - E\|^2_F + \|Z - J\|^2_F)$$ (10)

Since Problem (9) was unconstrained, we could take the derivative of $Z$ directly. We obtained the derivation result of Problem (10) as follows:

$$\frac{\partial L}{\partial Z} = -X^T PY_1 + Y_2 + \frac{\mu}{2} \left( -2X^T PX + 2X^T PP^T XZ + 2X^T PE + Z - J \right)$$ (11)

Let $\frac{\partial L}{\partial Z} = 0$, then we could obtain the result of $Z$ as follows:

$$Z^* = \left( X^T PP^T X + I \right)^{-1} \left( \frac{X^T PY_1 - Y_2}{\mu} + J + X^T P(X - E) \right)$$ (12)

3.3.3. Subproblem $E$

To update $E$, we needed to solve the following problem:

$$E^* = \arg\min_E \frac{\lambda}{\mu} \|E\|_{2,1} + \frac{1}{2} \|E - (X - P^T XZ + Y_1 / \mu)\|_F^2$$ (13)

As with Problem (9), Problem (13) also had a closed-form solution. We calculated $E$ using Lemma 1.

Lemma 1. Let $Q = [q_1, q_2, \ldots, q_i, \ldots]$ be a given matrix. When the optimal solution to

$$\min_W \lambda \|W\|_{2,1} + \frac{1}{2} \|W - Q\|_F^2$$ (14)

is $W^*$, then the $i$-th column of $W^*$ is

$$[W^*]_{i} = \begin{cases} \frac{\|q_i\|_2 - \lambda}{\|q_i\|_2} q_i & \text{if } \lambda < \|q_i\|_2 \\ 0 & \text{otherwise} \end{cases}$$ (15)

3.3.4. Subproblem $P$

To update $P$, we needed to solve the following problem:

$$P^* = \arg\min_P \text{tr} \left( Y_1^T (X - P^T XZ - E) \right) + \frac{\mu}{2} \|X - P^T XZ - E\|^2_F \text{ s.t. } P^T XX^T P = I_d$$ (16)

Considering that Problem (16) was a constrained problem, we introduced Lemma 2 to solve it.

Lemma 2. Given the objective function $\min_R \|Q - GR\|_F^2 \cdot \text{s.t. } R^T R = RR^T = I$, the optimal solution is $R = UV^T$, where $U$ and $V$ are the left and right singular values of the SVD decomposition of $G^T Q$, respectively.
We transformed Problem (16) to obtain the following results:

\[ P^* = \arg\min_P \frac{\mu}{2} ||X - P^T XZ - E + Y_1/\mu||_F^2 \quad \text{s.t.} \quad P^T XX^T P = I_d \]

Going one step further:

\[ P^* = \arg\min_P \frac{\mu}{2} ||(X + Y_1/\mu - E)^T - Z^T XZ P||_F^2 \quad \text{s.t.} \quad P^T XX^T P = I_d \]

Let \( X^T P = R \), then according to Lemma 2, we could obtain the equation \( X^T P = UV^T \). Then, we only needed to calculate the inverse of the data matrix to obtain the solution to Problem (16): \( P = (X^T)^{-1} UV^T \).

The detailed optimization algorithm for DLWECCL is shown in Algorithm 2.

**Algorithm 2: Optimization algorithm for DLWECCL.**

**Input:** Microcluster representation matrix \( X \), parameter \( \lambda \).

**Output:** Output: \( Z, P, E \).

**Initialization:** \( Z = J = 0, E = 0, P = I, Y_1 = 0, Y_2 = 0, \mu = 0.5, \max\mu = 10^6, \rho = 2.3 \) and \( \epsilon = 10^{-6} \).

\[ \textbf{while not converged do} \]

Update variables \( P, Z, E \) and \( J \) according to subproblems 1-4.

Update the multipliers:

\[ Y_1 = Y_1 + \mu(X - P^T XZ - E), \]

\[ Y_2 = Y_2 + \mu(Z - J), \]

Update the parameter \( \mu \) by \( \mu = \min(\rho \mu, \max\mu) \).

Check the convergence conditions: \( ||X - P^T XZ - E||_\infty < \epsilon \) and \( ||Z - J||_\infty < \epsilon \)

\[ \textbf{end} \]

3.3.5. Differences between Our Approach and Other Ensemble Clustering Methods

As mentioned in the Introduction, our method introduces the theory of microclusters in order to reduce the dataset size. The divergence weights are then calculated and applied to the microclusters. Finally, a low-rank representation is performed to obtain a similarity matrix. Compared to other existing advanced methods, our method has a great number of differences and advantages, mainly in the following aspects:

1. Differences in the data matrix. Some methods perform ensemble algorithms based on co-association (CA) matrices [10,27], but CA matrices focus on instance-level relationships and ignore the relationships between clusters. Our method is based on instance–cluster data matrices, although the DREC [13], PTA-CL [17] and CESHL [11] methods also use data matrices that are similar to ours. Among these methods, CESHL does not introduce microclusters and its time efficiency is low. DREC fails to consider the differences between microclusters. Our method makes up for these shortcomings. It is worth pointing out that although the PTA-CL method considers the differences between microclusters, it does not explore their deep structures.

2. Differences in the weighting methods. The LWEC method is based on the entropy-based weighting method [18]. As shown in Section 3.1, the entropy-based weighted method cannot solve the problem of consistent weights among the similar clusters. Therefore, our method uses KL divergence-based weighting to alleviate this contradiction to a certain extent. Some other weighting methods focus on cluster-level similarities and then map these similarities to the instance level [16].
(3) Differences in the low-rank representation. The existing low-rank representation-based ensemble methods all treat the original data directly as a dictionary [28,29]. Considering that good dictionaries are crucial to the learning of similarity matrices, our method uses novel low-rank representation with dictionary learning constraints.

4. Experiments

4.1. Datasets and Evaluation Methods

In this section, we present the setup and results of our extensive experiments to validate the proposed algorithm on 20 real datasets. Information about the datasets is listed in Table 2.

Table 2. The characteristics of the datasets.

| Dataset      | Instances | Features | Classes | Dataset | Instances | Features | Classes |
|--------------|-----------|----------|---------|---------|-----------|----------|---------|
| Zoo          | 101       | 16       | 7       | ISOLET  | 7797      | 617      | 26      |
| Control      | 600       | 60       | 6       | MNIST   | 5000      | 784      | 10      |
| Segment      | 2310      | 18       | 7       | OOR     | 5620      | 64       | 10      |
| MnistData_05 | 3495      | 653      | 10      | Semion  | 1593      | 256      | 10      |
| Binalpha     | 1404      | 320      | 36      | SPF     | 1941      | 27       | 7       |
| MnistData_10 | 6996      | 688      | 10      | Texture | 5500      | 40       | 11      |
| Caltech101   | 8671      | 784      | 10      | VS      | 846       | 18       | 4       |
| Caltech20    | 2386      | 30,000   | 20      | Wine    | 178       | 13       | 3       |
| FCT          | 3780      | 54       | 7       | MF      | 2000      | 649      | 10      |
| IS           | 2310      | 19       | 7       | LS      | 6435      | 36       | 6       |

Although there are various metrics for evaluating clustering performance, we chose three of them, namely accuracy (ACC), normalized mutual information (NMI) and adjusted rand index (ARI), to evaluate the proposed approach because of their simplicity, popularity and robustness to changes in labeling [18,30].

ACC is the score that is obtained by matching ground truth labels. Since the labels that are assigned by clustering methods may be inconsistent with the ground truth labels, the Hungarian algorithm is generally used for label alignment when calculating ACC, which can be formulated as follows:

\[
\text{ACC} = \max_f \frac{1}{n} \sum_{j=1}^{n} \delta(y_j, f(\pi(x_j))),
\]

where \(y_j\) represents the ground truth labels and \(\delta(y_j, f(\pi(x_j))) = 1\) when \(y_j = f(\pi(x_j))\) and \(\delta(y_j, f(\pi(x_j))) = 0\) otherwise.

As a measure of mutual information entropy that indicates the clustering results and the ground truth labels [31], NMI is defined as follows:

\[
\text{NMI} = \frac{\sum_p \sum_q n_{p,q} \log \left(\frac{n_{p,q}}{n_p n_q}\right)}{\sqrt{\sum_p n_p \log \frac{n_p}{N} \sum_q n_q \log \frac{n_q}{N}}},
\]

where the cluster \(c_p\) in the clustering results and the cluster \(c_q\) in the ground truth labels contain \(n_p\) and \(n_q\) instances, respectively.

ARI is an improved version of the rand index (RI) that can reflect the degree of overlap between clustering results and ground truth labels [32], which can be defined as follows:

\[
\text{ARI} = \frac{1}{\binom{N}{2}} \left[ \binom{N_k}{2} \sum_{i=1}^{k} \binom{N_{l_i}}{2} + \binom{N_k'}{2} \sum_{j=1}^{k'} \binom{N_{l_j}}{2} \right] - \left[ \binom{N_k}{2} \sum_{i=1}^{k} \binom{N_{c_i}}{2} + \binom{N_k'}{2} \sum_{j=1}^{k'} \binom{N_{c_j}}{2} \right],
\]

where the clustering results and the ground truth labels contain \(k\) and \(k'\) clusters, respectively, \(N_{l_{ij}}\) is the number of common instances in cluster \(c_i\) in the clustering results and
cluster $p_j$ in the ground truth labels and $N_c^j$ and $N_p^j$ are the numbers of instances in clusters $c_i$ and $p_j$, respectively.

The definitions of these three evaluation indicators show that the greater the indicator values, the better the method.

4.2. Experimental Settings

Each of the selected datasets contained 100 base clustering results, from which we randomly selected 20 to evaluate the ensemble clustering in each run. There were two main hyperparameters in the proposed approach, namely $\theta$ in (3) and $\lambda$ in Problem (7). We used the grid search method to optimize the hyperparameters with all of the data in each dataset using the set of $\{0.2 : 0.1 : 2\}$ for $\theta$ and $\{0.01, 0.1, 1, 10, 100, 200, 500\}$ for $\lambda$. Note that these hyperparameters could also be optimized using evolutionary algorithms, as in many practical applications [33–36]. Additionally, the true number of classes in each dataset was also used as the input of the proposed approach. For each dataset, we ran the experiments 20 times and then reported the average results.

4.3. Experimental Results

We carried out a large number of repeated experiments and obtained average results, according to the optimal parameter range. We also compared our method to the following models:

- **DREC** [13], which introduces microclusters to reduce the amount of data and is a dense representation-based method;
- **LWGP, LWEA** [18], which both use locally weighted methods (LWGP is based on graph partitioning and LWEA is based on hierarchical clustering);
- **MCLA** [37], which is a clustering ensemble method that is based on hypergraph partitioning;
- **PTA-CL** [17], which introduces microclusters, explores probabilistic trajectories based on random walks and then uses complete-linkage hierarchical agglomerative clustering;
- **CESHL** [11], which is a clustering ensemble method for structured hypergraph learning;
- **SPCE** [10], which introduces a self-paced learning method to learn consensus results from base clusterings;
- **TRCE** [27], which is a multi-graph learning clustering ensemble method that considers tri-level robustness.

Note that the proposed DLWECDL used divergence-based local weights for ensemble clustering. We also replaced the divergence-based local weights in DLWECDL with entropy-based local weights but kept the other components unchanged in another algorithm, called ELWECDL, for comparison.

The NMI values of the proposed DLWECDL method and the other selected methods are listed in Table 3, where the best and second best values are shown in bold. From this table, it can be seen that the proposed DLWECDL method achieved the best or second best result in 16 out of the 20 cases, followed by the SPCE and ELWECDL methods (best or second best in 8 out of the 20 cases). TRCE achieved the best or second best result three times, meaning it ranked fourth among the ten methods. DREC, LWGP, LWEA, PTA-CL and CESHL performed so poorly that they all only achieved the best or second best result once. MCLA did not achieve the best or second best value for any of the 20 datasets. On average, DLWECDL and ELWECDL improved the NMI values by 7.36% and 6.12%, respectively, compared to the other eight ensemble clustering models. Therefore, these results demonstrated that the proposed DLWECDL significantly outperformed the other selected ensemble clustering methods in terms of NMI.
The ARI values of the ensemble clustering methods that are shown in Table 4 offered the following findings: (1) the DLWECDL method achieved the best or second best results 17 times, meaning that it ranked first among all of the ensemble clustering methods once again; (2) the DLWECDL method was followed by the ELWECDL method, which achieved the best or second best results for nine datasets; (3) the rest of the methods only achieved the best or second best values three times or less and specially, both MCLA and TRCE failed to achieve the best or second best values for any of the datasets; (4) on average, the ARI values of DLWECDL and ELWECDL improved by 15.11% and 12.49%, respectively, compared to the other models. These findings confirmed that the proposed DLWECDL method was superior to the other selected methods in terms of ARI.

Table 4. Our comparison of the proposed method to the other selected methods, according to ARI.

| Dataset | DREC [13] | LWGP [18] | DWEA [16] | MCLA [17] | PTA-CL [17] | CESHL [11] | SPCE [10] | TRCE [27] | ELWECDL | DLWECDL |
|---------|-----------|-----------|-----------|------------|-------------|-------------|-----------|-----------|----------|----------|
| Wine    | 0.7522    | 0.7827    | 0.7630    | N/A        | N/A         | 0.7053      | 0.7645    | 0.7688    | 0.7978   | 0.7892   |
| IS      | 0.4223    | 0.4570    | 0.4228    | 0.6225     | 0.6225      | 0.6042      | 0.6592    | 0.6322    | 0.6875   | 0.6682   |
| Binalpha| 0.5888    | 0.5922    | 0.5977    | 0.5824     | 0.5439      | 0.5690      | 0.5965    | 0.5965    | 0.6068   | 0.5965   |
| Caltech20| 0.5467    | 0.5327    | N/A       | 0.5221     | 0.5599      | N/A         | N/A       | N/A       | 0.5486   | 0.5519   |
| Caltech10| 0.4205    | 0.4380    | 0.4520    | 0.3844     | 0.4485      | 0.4600      | 0.4690    | 0.4690    | 0.4940   | 0.4830   |
| Minst_DATA_05| 0.5016    | 0.4517    | 0.4607    | 0.4876     | 0.5004      | 0.4963      | 0.4898    | 0.4898    | 0.5320   | 0.5200   |
| 0.5016  | 0.4517    | 0.4607    | 0.4876    | 0.5004      | 0.4963      | 0.4898      | 0.4898    | 0.4898    | 0.5320   | 0.5200   |
| ZOO     | 0.6312    | 0.5848    | 0.6026    | 0.7960     | 0.7735      | 0.5891      | 0.5974    | 0.6065    | 0.6065   | 0.6065   |
| Segment | 0.5872    | 0.5889    | 0.5900    | 0.5944      | 0.5994      | 0.6051      | 0.6096    | 0.6096    | 0.6096   | 0.6096   |

We further ran DREC, ELWECDL and DLWECDL on eight datasets (Wine, Caltech20, Caltech10, Control, FCT, ISOLET, LS and SPF). Each ensemble clustering method was run 20 times on each dataset and the accuracy values are plotted in Figure 5. We found that the ELWECDL and DLWECDL methods achieved much higher accuracy than the DREC method in almost all cases. Meanwhile, the DLWECDL method was advantageous over the ELWECDL method in most cases, which indicated that the divergence-based local weights were better than the entropy-based local weights for ensemble clustering.
4.4. Impact of Hyperparameters

For the proposed ensemble clustering algorithm, there are two main hyperparameters, i.e., $\lambda$ in Problem (7) and $\theta$ in ECI. According to our extensive experiments, we found that
λ had little effect on the final clustering results. The reason for this is the fact that low-rank structures are mainly explored using low-rank subspace clustering methods and \( \|Z\| \), dominated Problem (7), as confirmed by Chen et al. [25]. For the weight parameter \( \theta \), we found that it had a large influence on the final results and that the optimal value of \( \theta \) was related to the random selection of base clusterings in each experiment. According to our experience, the optimal weight parameter was 0.2–2. We selected some other datasets and repeated the experiments another 50 times. The \( \theta \) values that corresponded to the maximum NMI values are shown in Figure 6.

![Figure 6](image)

Figure 6. The weight parameter change diagrams that corresponded to the optimal results of a single repeated experiment.

As shown in Figure 6a, in the first run of the experiment on the Zoo dataset, the corresponding optimal \( \theta \) value was 1.2, which became 1.3 in the second run. In the subsequent experiment runs, \( \theta \) did not have a fixed optimal value. The other datasets showed this same trend, which indicated that the parameter \( \theta \) in our method was associated with the data matrix, i.e., we could not fix the weight parameter \( \theta \), even within the same dataset. This was mainly due to the problem of base clustering set selection.

### 4.5. Running Time

We compared the running time of the selected algorithms on 10 datasets, as shown in Table 5. As can be seen from the table, the time efficiency of the DLWECDL algorithm was not good because many iterations were performed while looking for low-rank repre-
sensation. In order to reduce the number of iterations, we could adjust the learning rate, i.e., $\rho$, within an appropriate range, as long as the loss function was reasonably reduced. By adjusting the $\rho$ value, we could control the number of iterations at less than 10, thereby improving the time performance of the algorithm. As can be seen from the table, after we increased the $\rho$ value, the running time of DLWECDL became less than that of DREC [13].

|                      | HGBF [38] | SEC [39] | PTGP [17] | PTA-AL [17] | MCLA [17] | LWGP [18] | DREC [13] | DLWECDL ($\rho = 2.3$) | DLWECDL ($\rho = 2.3$) |
|----------------------|-----------|----------|-----------|-------------|-----------|-----------|-----------|----------------------|----------------------|
| Caltech20            | 1.2448    | 0.2952   | 0.1626    | 0.1739      | 0.889     | 0.9445    | 11.4153   | 26.522               | 26.679               |
| FCT                  | 0.3354    | 23.4457  | 0.0773    | 0.0211      | 0.8295    | 0.1336    | 23.6359   | 26.7346              | 23.344               |
| IS                   | 0.2218    | 5.4817   | 0.0146    | 0.012       | 0.4531    | 0.1062    | 0.8407    | 5.5264               | 1.7756               |
| ISOLET               | 0.9808    | 151.9921 | 0.1063    | 0.0175      | 1.0064    | 0.1419    | 82.7747   | 123.2756             | 56.522               |
| MNIST                | 0.3202    | 41.2678  | 0.1686    | 0.0231      | 0.8589    | 0.0556    | 53.722    | 129.5073             | 74.014               |
| ODR                  | 0.3156    | 56.0393  | 0.0776    | 0.0175      | 0.8576    | 0.0549    | 51.6894   | 51.0303              | 26.0803              |
| SPY                  | 0.0776    | 3.2589   | 0.0235    | 0.0072      | 0.4514    | 0.0284    | 0.8558    | 1.1861               | 0.4535               |
| Semeion              | 0.0067    | 2.7935   | 0.0064    | 0.0116      | 0.4649    | 0.0429    | 3.2473    | 4.2067               | 2.6795               |
| Texture              | 0.2214    | 63.5031  | 0.0514    | 0.0136      | 0.7044    | 0.0703    | 9.3535    | 29.9032              | 20.1014              |
| Vi                   | 0.0822    | 0.4591   | 0.0242    | 0.006       | 0.6647    | 0.0228    | 0.6316    | 1.2248               | 0.7616               |

4.6. Discussion

As analyzed in Section 3.3.5, our method is different from the other selected ensemble clustering methods in several aspects. Among them, the CESHL, DREC, PTA-CL and PTGP methods are based on the same data matrix as ours, while TRCE and SPCE are based on CA matrices. DREC, PTA-CL and PTGP all introduce microclusters to reduce the amount of data, while CESHL uses all data matrices directly. The superiority of our method over these methods mainly stems from the idea of the weighting and low-rank representation methods.

The KL divergence-based weighting method measures the differences between clusters, which alleviates the problem of the significant weight differences between similar clusters in ELWEC. Currently, DREC treats all microclusters equally and fails to consider the differences between microclusters. Although PTA-CL, PTGP and CESHL consider the differences between microclusters or clusters, none of them apply low-rank representation, i.e., they offer an insufficient exploration of the underlying information in data matrices. Moreover, CESHL is limited by the scale of the data, which leads to lower time efficiency.

Clustering ensemble method based on low-rank representation, such as RSEC, NRSEC, etc., are based on CA matrices and focus on instance-level relationships. They also all use the original data directly, i.e., the CA matrices, as dictionaries, although the $L_{2,1}$-norm is applied to consider the influence of noise. In general, the advantages of dictionary learning are more obvious.

5. Conclusions

In this paper, we proposed a new weighting method and a new low-rank representation method with adaptive dictionary learning. The new weighting method was able to mine more effective cluster–cluster relationships. We mapped these inter-cluster relationships into a representative microcluster matrix, i.e., we used the microcluster–cluster matrix as a new data matrix, and added new effective information on the basis of retaining the original matrix information to the greatest possible extent. Furthermore, methods based on low-rank representation with adaptive dictionary learning have been shown to be effective and we used a more reasonable $L_{2,1}$-norm to enhance robustness. Our experimental results demonstrated the effectiveness of our proposed method. On average, the proposed DLWECDL improved the NMI and ARI values by 7.36% and 15.11%, respectively, compared to the other selected SOTA ensemble clustering models. However, due to the influence of the random selection of base clusterings, we could not obtain a fixed optimal weight parameter that matched all possible base clustering combinations, even within the same dataset. Through our extensive experiments, we obtained an empirical range of weight parameters. The selection of the optimal combination of base clusterings within a dataset to obtain a pre-determined optimal weight parameter is our next research direction.
Author Contributions: Conceptualization, J.X. and T.L.; formal analysis, J.W. and T.L.; investigation, J.X. and Y.N.; methodology, J.X. and T.L.; project administration, T.L.; resources, T.L.; software, J.X.; supervision, T.L.; validation, J.X.; writing—original draft preparation, J.X., J.W. and T.L.; writing—review and editing, J.X., J.W., T.L. and Y.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Education of Humanities and Social Science Project (grant no. 19YJAZH047), the Scientific Research Fund of the Sichuan Provincial Education Department (grant no. 17ZB0433) and the Key Entrusted Projects of Higher Education Teaching Reform in Heilongjiang Province (grant no. SJGZ20200067).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the data in this paper are publicly available. They can be accessed at http://archive.ics.uci.edu/ml, http://www.cs.nyu.edu/~roweis/data.html, and https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html (all accessed on 10 April 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Zhou, Z.H. Machine Learning; Springer Nature: Berlin/Heidelberg, Germany, 2021.
2. Rupp, A.A. Clustering and Classification; Oxford University Press: Oxford, UK, 2013.
3. Omran, M.G.; Engelbrecht, A.P.; Salman, A. An overview of clustering methods. Intell. Data Anal. 2007, 11, 583–605. [CrossRef]
4. Li, T.; Qian, Z.; Deng, W.; Zang, D.; Lu, H.; Wang, S. Forecasting crude oil prices based on variational mode decomposition and random sparse Bayesian learning. Appl. Soft Comput. 2021, 113, 108032. [CrossRef]
5. Saxena, A.; Prasad, M.; Gupta, A.; Bharill, N.; Patel, O.P.; Tiwari, A.; Er, M.J.; Ding, W.; Lin, C.T. A review of clustering techniques and developments. Neurocomputing 2017, 267, 664–681. [CrossRef]
6. Mittal, M.; Goyal, L.M.; Hemanth, D.J.; Sethi, J.K. Clustering approaches for high-dimensional databases: A review. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 2019, 9, e1300. [CrossRef]
7. Golalipour, K.; Akbari, E.; Hamidi, S.S.; Lee, M.; Enayatifar, R. From clustering to clustering ensemble selection: A review. Eng. Appl. Artif. Intell. 2021, 104, 104388. [CrossRef]
8. Zhang, M. Weighted clustering ensemble: A review. Pattern Recognit. 2021, 124, 108428. [CrossRef]
9. Wu, X.; Ma, T.; Cao, J.; Tian, Y.; Alabdulkarim, A. A comparative study of clustering ensemble algorithms. Comput. Electr. Eng. 2018, 68, 603–615. [CrossRef]
10. Zhou, P.; Du, L.; Liu, X.; Shen, Y.D.; Fan, M.; Li, X. Self-paced clustering ensemble. IEEE Trans. Neural Netw. Learn. Syst. 2020, 32, 1497–1511. [CrossRef]
11. Zhou, P.; Wang, X.; Du, L.; Li, X. Clustering ensemble via structured hypergraph learning. Inf. Fusion 2022, 78, 171–179. [CrossRef]
12. Huang, D.; Wang, C.D.; Wu, J.S.; Lai, J.H.; Kwoh, C.K. Ultra-scalable spectral clustering and ensemble clustering. IEEE Trans. Knowl. Data Eng. 2019, 32, 1212–1226. [CrossRef]
13. Zhou, J.; Zheng, H.; Pan, L. Ensemble clustering based on dense representation. Neurocomputing 2019, 357, 66–76. [CrossRef]
14. Li, F.; Qian, Y.; Wang, J.; Dang, C.; Jing, L. Clustering ensemble based on sample’s stability. Artif. Intell. 2019, 273, 37–55. [CrossRef]
15. Jia, Y.; Tao, S.; Wang, R.; Wang, Y. Ensemble Clustering via Co-association Matrix Self-enhancement. arXiv 2022, arXiv:2205.05937.
16. Huang, D.; Wang, C.D.; Peng, H.; Lai, J.; Kwoh, C.K. Enhanced ensemble clustering via fast propagation of cluster-wise similarities. IEEE Trans. Syst. Man Cybern. Syst. 2018, 51, 508–520. [CrossRef]
17. Huang, D.; Lai, J.H.; Wang, C.D. Robust ensemble clustering using probability trajectories. IEEE Trans. Knowl. Data Eng. 2015, 28, 1312–1326. [CrossRef]
18. Huang, D.; Wang, C.D.; Lai, J.H. Locally weighted ensemble clustering. IEEE Trans. Cybern. 2017, 48, 1460–1473. [CrossRef]
19. Wang, L.; Luo, J.; Wang, H.; Li, T. Markov clustering ensemble. Knowl. Based Syst. 2022, 251, 109196. [CrossRef]
20. Li, F.; Qian, Y.; Wang, J. GoT: A Growing Tree Model for Clustering Ensemble. In Proceedings of the AAAI Conference on Artificial Intelligence; Published by AAAI Press: Palo Alto, CA, USA, 2021; Volume 35, pp. 8349–8356.
21. Jia, Y.; Liu, H.; Hou, J.; Zhang, Q. Clustering ensemble meets low-rank tensor approximation. In Proceedings of the AAAI Conference on Artificial Intelligence; Published by AAAI Press: Palo Alto, CA, USA, 2021; Volume 35, pp. 7970–7978.
22. Ji, X.; Liu, S.; Yang, L.; Ye, W.; Zhao, P. Clustering ensemble based on approximate accuracy of the equivalence granularity. Appl. Soft Comput. 2022, 129, 109492. [CrossRef]
23. Akbari, E.; Dahlan, H.M.; Ibrahim, R.; Alizadeh, H. Hierarchical cluster ensemble selection. Eng. Appl. Artif. Intell. 2015, 39, 146–156. [CrossRef]
24. Jia, J.; Xiao, X.; Liu, B.; Jiao, L. Bagging-based spectral clustering ensemble selection. *Pattern Recognit. Lett.* 2011, 32, 1456–1467. [CrossRef]
25. Chen, J.; Mao, H.; Wang, Z.; Zhang, X. Low-rank representation with adaptive dictionary learning for subspace clustering. *Knowl. Based Syst.* 2021, 223, 107053. [CrossRef]
26. Lin, Z.; Liu, R.; Su, Z. Linearized alternating direction method with adaptive penalty for low-rank representation. *Adv. Neural. Inf. Process. Syst.* 2011, 24, 1–9.
27. Zhou, P.; Du, L.; Shen, Y.D.; Li, X. Tri-level robust clustering ensemble with multiple graph learning. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*; Published by AAAI Press: Palo Alto, CA, USA 2021, pp. 11125–11133.
28. Tao, Z.; Liu, H.; Li, S.; Ding, Z.; Fu, Y. Robust spectral ensemble clustering via rank minimization. *ACM Trans. Knowl. Discov. Data (TKDD)* 2019, 13, 1–25. [CrossRef]
29. Tao, Z.; Liu, H.; Li, S.; Fu, Y. Robust spectral ensemble clustering. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 24–28 October 2016; pp. 367–376.
30. Jing, L.; Tian, K.; Huang, J.Z. Stratified feature sampling method for ensemble clustering of high dimensional data. *Pattern Recognit.* 2015, 48, 3688–3702. [CrossRef]
31. Shao, M.; Li, S.; Ding, Z.; Fu, Y. Deep linear coding for fast graph clustering. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
32. Hubert, L.; Arabie, P. Comparing partitions. *J. Classif.* 1985, 2, 193–218. [CrossRef]
33. Li, T.; Qian, Z.; He, T. Short-term load forecasting with improved CEEMDAN and GWO-based multiple kernel ELM. *Complexity* 2020, 2020. [CrossRef]
34. Li, T.; Shi, J.; Deng, W.; Hu, Z. Pyramid particle swarm optimization with novel strategies of competition and cooperation. *Appl. Soft Comput.* 2022, 121, 108731. [CrossRef]
35. Deng, W.; Ni, H.; Liu, Y.; Chen, H.; Zhao, H. An adaptive differential evolution algorithm based on belief space and generalized opposition-based learning for resource allocation. *Appl. Soft Comput.* 2022, 127, 109419. [CrossRef]
36. Li, T.; Shi, J.; Zhang, D. Color image encryption based on joint permutation and diffusion. *J. Electron. Imaging* 2021, 30, 013008. [CrossRef]
37. Strehl, A.; Ghosh, J. Cluster ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* 2002, 3, 583–617.
38. Fern, X.Z.; Brodley, C.E. Solving cluster ensemble problems by bipartite graph partitioning. In Proceedings of the Twenty-First International Conference on Machine Learning, Banff, Alberta, Canada, 4 July 2004; p. 36.
39. Liu, H.; Wu, J.; Liu, T.; Tao, D.; Fu, Y. Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence. *IEEE Trans. Knowl. Data Eng.* 2017, 29, 1129–1143. [CrossRef]