

Article

Local-Sample-Weighted Clustering Ensemble with High-Order Graph Diffusion

Jianwen Gan ^{1,2,†} , Yunhui Liang ^{1,2,†} and Liang Du ^{1,2,*} ¹ Institute of Big Data Science and Industry, Shanxi University, Taiyuan 030031, China² School of Computer and Information Technology, Shanxi University, Taiyuan 030031, China

* Correspondence: duliang@sxu.edu.cn

† These authors contributed equally to this work.

Abstract: The clustering ensemble method has attracted much attention because it can improve the stability and robustness of single clustering methods. Among them, similarity-matrix-based methods or graph-based methods have had a wide range of applications in recent years. Most similarity-matrix-based methods calculate fully connected pairwise similarities by treating a base cluster as a whole and ignoring the importance of the relevance ranking of samples within the same base cluster. Since unreliable similarity estimates degrade clustering performance, constructing accurate similarity matrices is of great importance in applications. Higher-order graph diffusion based on reliable similarity matrices can further uncover potential connections between data. In this paper, we propose a more substantial graph-learning-based ensemble algorithm for local-sample-weighted clustering, which implicitly optimizes the adaptive weights of different neighborhoods based on the ranking importance of different neighbors. By further diffusion on the consensus matrix, we obtained an optimal consistency matrix with more substantial discriminative power, revealing the potential similarity relationship between samples. The experimental results showed that, compared with the second-best DREC algorithm, the accuracy of the proposed algorithm improved by 17.7%, and that of the normalized mutual information (NMI) algorithm improved by 15.88%. All empirical results showed that our clustering model consistently outperformed the related clustering methods.

Keywords: clustering ensemble; localized weighted; sparse; graph diffusion**MSC:** 68-04

Citation: Gan, J.; Liang, Y.; Du, L. Local-Sample-Weighted Clustering Ensemble with High-Order Graph Diffusion. *Mathematics* **2023**, *11*, 1340. <https://doi.org/10.3390/math11061340>

Academic Editor: Zhao Kang

Received: 17 January 2023

Revised: 2 March 2023

Accepted: 6 March 2023

Published: 9 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Data clustering is an essential unsupervised learning technique widely studied in different fields, such as statistics, pattern recognition, machine learning, and data mining. Clustering divides data objects into separate subsets, each of which is called a cluster. An effective clustering process makes the objects in a cluster as similar as possible, but as dissimilar as possible from the objects in other clusters. Although significant progress has been made in knowledge discovery, traditional machine learning methods still need to achieve satisfactory performance when dealing with complex data such as unbalanced data, high-dimensional data, data with noise, etc. These methods have difficulty capturing information, such as various features and the underlying structure of the data.

The clustering ensemble method provides a framework for combining multiple weak data set clusters to generate a consensus clustering [1]. As a research hotspot, the clustering ensemble method aims to integrate data fusion, modeling, and mining into a unified framework [2]. Specifically, the clustering ensemble method first generates several sets of consensus information through different mechanisms. Based on these learned features, multiple consensus learning algorithms are applied to create preliminary prediction results with low confidence. Finally, consensus learning can fuse the knowledge with enough information to achieve knowledge discovery and obtain better prediction performance

using constrained source information. The clustering ensemble method uses the consensus learning technique to find a new data division based on multiple clustering results, which shares the clustering information of all the input clustering results on the data set to the greatest extent. The clustering ensemble method can effectively improve the robustness and stability of a single cluster [3]. The clustering ensemble technique has high parallelism and expansibility [4].

Over the past few decades, researchers have proposed many clustering ensemble algorithms. Among these methods, one popular method in recent years has been the similarity-matrix-based clustering ensemble method. The similarity-matrix-based clustering ensemble method is easy to implement and has the advantage of being a simple concept [5]. In addition, the consensus matrix provides a reasonable basis for subsequent analysis. Similarity-matrix-based methods [6–9] transform the information provided by the base clustering into a similarity matrix and then obtain the final clustering ensemble results by various methods such as agglomerative clustering and invoking partitioning. Although showing improved performance, exploring and exploiting local structures within these similarity-matrix-based clustering ensemble methods still encounter the following two critical issues:

- Existing methods based on the similarity matrix take advantage of the information of all sample relationships. However, in the clustering ensemble method, the quality of the base clusters plays a crucial role in the clustering process. Low-quality base clusters may seriously affect the consensus results [10]. This approach ignores the ranking importance of sample relevance and may suffer from unreliable proximity relationships, which degrade the clustering performance [11].
- Although similarity-based methods have shown exemplary performance in many applications, consensus matrix structures may sometimes fail to capture complex higher-order correlations between instances. In clustering ensemble tasks, the relationships between samples may be too challenging to describe, as different basic clustering methods may find other data structures [8,12].

To solve these problems, we propose a new structured similarity matrix learning clustering ensemble method with graph diffusion based on the learned structured similarity matrix in this paper. Unlike the traditional similarity-matrix-based clustering ensemble method, we propose a new local sample weighting model, i.e., different samples are ranked differently in importance for similarity. We first learned a consensus affinity matrix among multiple clustering results to reveal the underlying local structure. At the same time, we imposed a rank constraint on the Laplacian matrix of the learned similarity matrix to assign the ideal nearest neighbors so that the connected components in the data are accurate with respect to the number of clusters, with each connected component corresponding to one cluster. The model learns the data similarity matrix and the clustering structure to obtain the best clustering results [13]. The underlying flow structure is ignored in the direct calculation of the metric sample pair similarity [14]. We saw that only one type of similarity could not fully reveal the intrinsic relationship between objects. Therefore, we extended the analysis to consider higher-order relationships recursively [15]. Meanwhile, we also took the ranking importance of the sample relevance by using a generalized form of sparsified graph diffusion. The main contributions are summarized as follows:

- We explored the optimal affinity by the local sample reconstruction coefficient, which depends on its similarity matrix and the correlation of its neighbors to improve the discrimination and alleviate the redundancy of highly similar samples within a local clique.
- We exploited the affinity matrix by global heat kernel diffusion, which aggregates tight local cliques and extracts high-order similarity to consolidate the connection and reduce the noise across local cliques.
- We integrated the exploration of the local structure and global structure into successive steps; the effectiveness of each step can be verified, and the efficiency was improved compared to the common learning method.

2. Related Work

The clustering ensemble method aims to integrate multiple candidate weak clustering results to obtain a better result. Strehl and Ghosh proposed the concept of the clustering ensemble for the first time in [4]. According to the data representation form of the base cluster and the final clustering result generation algorithm, clustering ensemble methods can be roughly divided into the following four categories. (1) Label-alignment-based methods: These methods first realize the mutual mapping between the cluster from different base clusterings by solving the label-matching problem of the cluster and then combine multiple base clusterings on the aligned cluster. Representative algorithms include the alternative vote method based on mutual information proposed by Zhou et al. [16]. Li et al. [17] proposed the selective fusion method based on evidence theory. (2) Sample cluster binary-relation-based methods: These methods sequentially splice the results of different base clusters to obtain a new sample cluster relation matrix and use different strategies to enhance the sparse matrix and carry out ensemble learning. The related studies proposed enhancing the sample cluster relationship based on the link cluster relationship and the utility function based on entropy [18–20]. Fern et al. extended the graph cut method to the clustering ensemble method [21]. Li et al. proposed a method based on non-negative matrix decomposition [7]. (3) Graph-division-based methods: The base cluster set contains different objects, such as samples and clusters, and their relationships. Strehl et al. [4] proposed three different types of hypergraph models and ensemble algorithms, namely the Meta Clustering Algorithm (MCLA), the Cluster-based Similarity Partitioning Algorithm (CSPA), and the Hypergraph Partitioning Algorithm (HGPA). Huang et al. [22] used bipartite graphs based on the relationship between samples and clusters. (4) Consistent common-matrix-partition-based methods: A single base cluster can construct the co-association matrix between samples, and the consistent co-association matrix can be obtained by selecting, weighting, merging, enhancing, and other steps. The final integration result can be obtained by dividing the matrix. The main methods include (4.1) the spectral method based on a common consensus matrix. Yu et al. [23] proposed using the spectral method to carry out ensemble learning from the consistent co-linked matrix. Jia et al. [24] proposed combining the bagging-based base clustering selection method and spectral method for ensemble learning of subsets. Liu et al. [25] further studied the robustness and generalization of spectral methods in the clustering ensemble. (4.2) A hierarchical clustering method based on the consistent co-linking matrix: In the literature [6,26–28], the evidence accumulation method has been used to construct the similarity graph, and the hierarchical clustering algorithm has been used to generate the integration results. Wang et al. [29] proposed a fast hierarchical clustering algorithm based on the co-linked matrix tree structure. Zhong et al. [27] enhanced the sample similarity by using cluster tendency Visualization Analysis Technology (VAT). Hu et al. [28] proposed to use fuzzy distance and knowledge roughness in soft computing to characterize and enhance the similarity. Huang et al. [30] proposed the LWEA algorithm based on cluster uncertainty local weighting.

Although showing significant improvement in the final clustering result, most of these methods treat a base cluster as a whole and ignore the importance of sample correlation within the same base cluster. Unlike previous works, which did not simultaneously impose appropriate constraints to capture the underlying structure and weight of the similarity matrix, we learned a structured consensus matrix with local-sample-weighted enhancement. We further conducted high-order graph diffusion to enhance the discrimination ability by capturing the long-range connections among samples.

3. Proposed Method

This section details our proposed method and provides an efficient optimization solution. We first introduce some main notations in Table 1.

Table 1. Description of the main notations.

Notations	Descriptions
n	Number of instances.
m	Number of base clusterings.
c_i	Number of the i -th base clusters.
w_i	The weight of the i -th co-association matrix.
k	The number of neighbors.
$\mathbf{H}_i \in \mathcal{R}^{n \times c_i}$	The i -th incidence matrix of base clusterings.
$\mathbf{S}_i \in \mathcal{R}^{n \times n}$	The i -th pairwise co-association matrix.
$\mathbf{Z} \in \mathcal{R}^{n \times n}$	The consensus co-association matrix.
$\mathbf{D}_Z \in \mathcal{R}^{n \times n}$	The node degree matrix of \mathbf{Z} .
$\mathbf{L}_Z \in \mathcal{R}^{n \times n}$	The Laplacian matrix of \mathbf{Z} .
$\mathbf{P} \in \mathcal{R}^{n \times n}$	The symmetric transition matrix of \mathbf{Z} .

3.1. Local-Sample-Weighted Clustering Ensemble

Given a set of unlabeled samples $\mathcal{X} = \{x_1, \dots, x_n\}$, we can obtain a collection of base clusterings $\mathcal{H} = \{\mathbf{H}_1, \dots, \mathbf{H}_m\}$, where m is the number of base clusterings, $\mathbf{H}_i \in \mathcal{R}^{n \times c_i}$ denotes the result of the i -th clustering, and c_i denotes the number of clusters. Given \mathbf{H}_i , a pairwise co-association matrix can be equivalently expressed as $\mathbf{S}_i = \mathbf{H}_i \mathbf{H}_i^T \in \mathcal{R}^{n \times n}$, where each entry is either 1 if they are in the same cluster or 0 otherwise. Given a set of co-association matrices, we can search these candidates to obtain the consensus one by minimizing the aggregated reconstruction errors between them, which can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \sum_{i=1}^m \|\mathbf{S}_i - \mathbf{Z}\|_F^2 \\ \text{s.t.} \quad & \mathbf{Z} \geq 0, \mathbf{Z}\mathbf{1} = \mathbf{1}, \end{aligned} \quad (1)$$

where \mathbf{Z} is the consensus co-association matrix, $\mathbf{1} \in \mathcal{R}^{n \times 1}$ is a column vector with n elements where each entry is 1, $\mathbf{Z} \geq 0$ ensures the non-negative property, and $\mathbf{Z}\mathbf{1} = \mathbf{1}$ is used for row normalization. It can be seen that the larger the distance between sample pairs, the smaller the similarity. Equation (1) can be reformulated as

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \sum_{i=1}^m \text{tr}(\mathbf{S}_i \mathbf{S}_i^T - 2\mathbf{S}_i \mathbf{Z}^T + \mathbf{Z} \mathbf{Z}^T) \\ = \min_{\mathbf{Z}} \quad & \sum_{i=1}^m \text{tr}(\mathbf{S}_i \mathbf{S}_i^T) - 2 \sum_{i=1}^m \text{tr}(\mathbf{S}_i \mathbf{Z}^T) + \sum_{i=1}^m \text{tr}(\mathbf{Z} \mathbf{Z}^T) \\ = \min_{\mathbf{Z}} \quad & -2 \sum_{i=1}^m \text{tr}(\mathbf{S}_i \mathbf{Z}^T) + m \text{tr}(\mathbf{Z} \mathbf{Z}^T) \\ \text{s.t.} \quad & \mathbf{Z} \geq 0, \mathbf{Z}\mathbf{1} = \mathbf{1}. \end{aligned} \quad (2)$$

It can be seen that the first term captures the pairwise alignment between the candidate results and the consensus one, while the second term captures the uniform prior of \mathbf{Z} . To further improve the flexibility for consensus learning, we introduced a hyperparameter η to balance these two terms, which can be written as

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{w}} \quad & - \sum_{i=1}^m w_i \text{tr}(\mathbf{S}_i \mathbf{Z}^T) + \eta \|\mathbf{Z}\|_F^2 \\ \text{s.t.} \quad & \mathbf{Z} \geq 0, \mathbf{Z}\mathbf{1} = \mathbf{1}, \sum_{i=1}^m w_i^2 = 1, w_i \geq 0, \end{aligned} \quad (3)$$

where w_i is the weight of the i -th co-association matrix. The simplex constraint on \mathbf{w} is to avoid the trivial solution. It can be seen that Equation (3) leads to unnecessarily extremely sparse results when $\eta \rightarrow 0$, and Equation (3) leads to uniform and dense affinity when $\eta \rightarrow \infty$.

The introduction of η in the problem in Equation (3) also leads to two problems. First, the additional parameter η should be well-tuned to achieve good performance. However, tuning such parameters in an unsupervised task is not easy without the guidance of supervised information. Second, it is vitally important to characterize the underlying manifold structure within these local affinities, i.e., the consensus co-association matrix \mathbf{Z} . However, it is hard to keep the local structure for each row of \mathbf{Z} by such a global hyperparameter.

The optimization of \mathbf{Z} can be naturally decoupled into n subproblems, where only one row is associated at once. Based on such a structure of the optimization problem, we further addressed the above two issues by replacing the global parameter η with a set of local regularization parameters $\boldsymbol{\gamma} = (\sqrt{\gamma_1}, \sqrt{\gamma_2}, \dots, \sqrt{\gamma_n})$ and $\boldsymbol{\gamma} \in \mathcal{R}^{n \times 1}$, each of which is obtained from the corresponding row of \mathbf{Z} . Although there are more hyperparameters introduced by using $\boldsymbol{\gamma}$, the behavior of the local regularization parameter can be more easily analyzed on each row of \mathbf{Z} . The main purpose of γ_i is to control the sparsity of the i -th row of \mathbf{Z} . It is expected only to keep the top k nonzero entries and discard the others for each row of \mathbf{Z} . Such results could be obtained by setting the optimal value of γ_i . The values of these local regularization parameters can also be determined empirically without additional tuning, as illustrated in Section 4.4. Thus, the optimization problem can be written as

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{w}} \quad & - \sum_{i=1}^m w_i \text{tr}(\mathbf{Z}^T \mathbf{S}_i) + \|\mathbf{Q} \odot \mathbf{Z}\|_F^2 \\ \text{s.t.} \quad & \mathbf{Z} \geq 0, \mathbf{Z}\mathbf{1} = \mathbf{1}, \sum_{i=1}^m w_i^2 = 1, w_i \geq 0, \end{aligned} \quad (4)$$

where $\mathbf{Q} = \mathbf{1}\boldsymbol{\gamma}^T$ and \odot denotes the elementwise Hadamard product. Inspired by the recent work in structural similarity learning [13], we also required that the affinity matrix has exactly c connected components. This can be achieved by imposing the rank constraint on the induced Laplacian matrix. We then obtain the following problem:

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{w}} \quad & - \sum_{i=1}^m w_i \text{tr}(\mathbf{Z}^T \mathbf{S}_i) + \|\mathbf{Q} \odot \mathbf{Z}\|_F^2 \\ \text{s.t.} \quad & \mathbf{Z} \geq 0, \mathbf{Z}\mathbf{1} = \mathbf{1}, \sum_{i=1}^m w_i^2 = 1, w_i \geq 0, \text{rank}(\mathbf{L}_Z) = n - c, \end{aligned} \quad (5)$$

where $\mathbf{L}_Z = \mathbf{D}_Z - \frac{\mathbf{S}^T + \mathbf{S}}{2}$ is the Laplacian matrix, and the degree matrix $\mathbf{D}_Z \in \mathcal{R}^{n \times n}$ is defined as a diagonal matrix where the i -th diagonal element is $\sum_j (\mathbf{Z}_{ij} + \mathbf{Z}_{ji})/2$.

According to the above formula, the proposed method, Local-Sample-Weighted Clustering Ensemble (LSWCE), jointly optimizes a reasonable weight for each similarity matrix, consensus graph learning, and consensus neighborhood into a unified framework. LSWCE implements the implicit optimization of adaptive weights according to the sorting relation of different neighbors and corresponding samples. Instead of adjusting hyperparameter γ_i with a grid search, hyperparameter γ_i can be determined in advance. Consequently, the underlying local structure among these base clusterings can be well characterized by the consensus co-association matrix, where the clustering performance could also be improved by such local affinity learning.

3.2. Consensus Affinity Graph Smoothing via Graph Diffusion Convolution

After examining the discrepancy between the structured consensus matrix and the ground truth, there are two unsatisfying points [31,32]:

- The consensus matrix is not smooth. It can be easily verified that the consensus matrix \mathbf{Z} is sparse, as mentioned before. The coefficients between sample \mathbf{x}_i and most samples within the sample cluster were set to be 0. Such a weighting coefficient breaks the connections between these pairs, separates these samples into disjoint groups, and blocks the identification of the underlying cluster structure.
- The coefficient matrix is noisy. It is expected that samples should reconstruct the sample in the same cluster. Since the results of the base clusters are not accurate, the neighborhood relationship used for local reconstruction within all base clusterings is inevitably not accurate, as expected. For example, on the one hand, the sample \mathbf{x}_j in the top- k neighbors \mathcal{N}^i of a sample \mathbf{x}_i may not be in the same cluster as \mathbf{x}_i , leading to unexpected reconstruction coefficient $\mathbf{Z}_{ij} > 0$. On the other hand, many samples in the same cluster of \mathbf{x}_i are not used to reconstruct \mathbf{x}_i . As a result, the induced local reconstruction coefficient would be somewhat noisy.

We remedied the consensus matrix by resorting to the graph diffusion convolution, expected to obtain a smooth and denoised graph. The immediate information could be characterized by passing messages between neighboring samples. For example, given a symmetrical consensus affinity graph \mathbf{Z} , the diagonal degree matrix of nodes $\mathbf{D}_\mathbf{Z}$, the symmetric transition matrix $\mathbf{P} = \mathbf{D}_\mathbf{Z}^{-\frac{1}{2}} \mathbf{Z} \mathbf{D}_\mathbf{Z}^{-\frac{1}{2}}$, and the normalized graph Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{P}$, the second-order relationship between sample \mathbf{x}_i and sample \mathbf{x}_j can be obtained by the aggregation of the first-order neighbors in the form of $\mathbf{P}_{ij}^2 = \sum_{i'=1}^n \mathbf{P}_{ii'} \mathbf{P}_{i'j}$. We aimed to capture a larger range of immediate higher-order neighborhood information by recursively introducing a series of matrices $\mathbf{P}^2, \mathbf{P}^3, \dots, \mathbf{P}^\infty$ in the following form:

$$\mathbf{P}^2 = \mathbf{P}\mathbf{P} \quad (6)$$

$$\mathbf{P}^3 = \mathbf{P}^2\mathbf{P}. \quad (7)$$

We finally considered all exponents at once and obtained the following diffusion matrix.

$$\mathbf{G} = \sum_{t=0}^{\infty} w_t \mathbf{P}^t, \quad (8)$$

where w_t is the immediate weighting coefficient and $\mathbf{P}^0 = \mathbf{I}$ is the identity matrix where a sample is only connected to itself.

Choosing a different weight will affect the performance of the clustering results. We found that adopting the heat kernel is an effective choice [33].

$$w_t = e^{-\theta} \frac{\theta^t}{t!}, \quad (9)$$

where θ is a non-negative value that can be expressed as temperature. It can be verified that $\sum_{t=0}^{\infty} w_t = 1$, $w_t \in [0, 1]$. The heat kernel diffusion satisfies the heat equation and can be viewed as describing the heat flowing along the edge of the graph with time, where the Laplacian of the graph determines the flow rate. The final diffusion matrix can be written as

$$\begin{aligned} \mathbf{G} &= \sum_{t=0}^{\infty} w_t \mathbf{P}^t = e^{-\theta} \sum_{t=0}^{\infty} \frac{\theta^t}{t!} \mathbf{P}^t \\ &= \sum_{t=0}^{\infty} \frac{(-\theta)^t}{t!} (\mathbf{I} - \mathbf{P})^t = \exp^{-\theta \mathbf{L}}, \end{aligned} \quad (10)$$

where the last two equations hold according to the definition of the matrix exponential, as seen above; when t is small, the heat kernel is dominated by locally connected structures encoded in Laplacian \mathbf{L} . As t increases, the heat kernel is ruled by the global structure of the graph. θ is the diffusion parameter.

Instead of discretization on \mathbf{Z} directly, in the proposed method, Local-Sample-Weighted Clustering Ensemble with High-order graph diffusion (LSWCEH), we utilized it using the global heat kernel diffusion procedure, as shown in (8), and generated an enhancement graph. The heat kernel diffusion filters out the larger eigenvalues corresponding to the fine details, but is also noisy via exponential decay transformation while amplifying the smaller eigenvalues corresponding to the top clusters in the graph. The heat kernel diffusion smooths out the neighborhood by bridging the local and global structure via the aggregation of intermediate higher-order information. Consequently, the heat-kernel-diffusion-induced graph smoothing helps reduce undesirable distortions and noise while preserving important manifold structures.

4. Optimization and Analysis

We introduce how to optimize objective function Equation (5). We first need to handle the rank constraint. Since $\text{rank}(\mathbf{L}_Z) = n - c$, the c smallest eigenvalues of \mathbf{L}_Z should be zeros. To deal with this, we minimized $\sum_{i=1}^m \phi(\mathbf{L}_Z)$, where $\phi(\mathbf{L}_Z)$ denotes the i -th smallest eigenvalues of \mathbf{L}_Z . According to the Ky Fan theorem [34], by introducing an orthogonal auxiliary matrix $\mathbf{F} \in \mathcal{R}^{n \times c}$, we have $\sum_{i=1}^m \phi(\mathbf{L}_Z) = \min_{\mathbf{F}^T \mathbf{F} = \mathbf{I}} \text{tr}(\mathbf{F}^T \mathbf{L}_Z \mathbf{F})$. Therefore, Equation (5) can be rewritten as follows:

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{F}, \mathbf{w}} \quad & -\text{tr}\left(\sum_{i=1}^m w_i \mathbf{S}_i \mathbf{Z}^T\right) + \|\mathbf{Q} \odot \mathbf{Z}\|_F^2 + 2\lambda \text{tr}(\mathbf{F}^T \mathbf{L}_Z \mathbf{F}) \\ \text{s.t.} \quad & \mathbf{Z} \geq 0, \mathbf{Z}\mathbf{1} = \mathbf{1}, \sum_{i=1}^m w_i^2 = 1, w_i \geq 0, \\ & \mathbf{F}^T \mathbf{F} = \mathbf{I}, \end{aligned} \quad (11)$$

where λ is a self-adaptive parameter; as the rank changes, λ is updated automatically and should be large enough to control the rank of \mathbf{L}_Z . We initialized $\lambda = 1e^{-4}$ and adjusted it automatically during the iterations, i.e., doubled it if $\text{rank}(\mathbf{L}_Z) > n - c$ or halved it if $\text{rank}(\mathbf{L}_Z) < n - c$. Since the optimization objective is not convex, the original problem was separated into three subproblems.

4.1. Update with Respect to \mathbf{w}

Given \mathbf{Z} and \mathbf{F} , the rest of the subproblem with respect to \mathbf{w} becomes

$$\begin{aligned} \max_{\mathbf{w}} \quad & \sum_{i=1}^m w_i \delta_i \\ \text{s.t.} \quad & \sum_{i=1}^m w_i^2 = 1, w_i \geq 0, \end{aligned} \quad (12)$$

where $\delta_i = \text{tr}(\mathbf{S}_i \mathbf{Z}^T)$. This problem could be easily solved with a closed-form solution as follows:

$$w_i = \frac{\delta_i}{\sqrt{\sum_{i=1}^m \delta_i^2}}. \quad (13)$$

4.2. Optimization w.r.t. \mathbf{Z}

Given \mathbf{F} and \mathbf{w} , the rest of the subproblem with respect to w_i and (11) is transformed to n subproblems, and each one can be independently solved. Let \mathbf{b}_i denote the i -th row of \mathbf{Z} , and each subproblem can be solved by:

$$\begin{aligned} \min_{\mathbf{b}_i} \quad & \gamma_i \mathbf{b}_i \mathbf{b}_i^T + (2\lambda \mathbf{d}_i - \sum_{p=1}^m w_p \mathbf{S}_{p[i,:]}) \mathbf{b}_i \\ \text{s.t.} \quad & \mathbf{b}_i \mathbf{1} = 1, \mathbf{b}_i \geq 0, \end{aligned} \quad (14)$$

where $\mathbf{S}_{p[i,:]}$ denotes the i -th row of the p -th base clustering. \mathbf{d}_i denotes the i -th rows of the \mathbf{Z} Euclidean distance matrix squared. Furthermore, (14) can be rewritten as a Quadratic Programming (QP) problem:

$$\begin{aligned} \min_{\mathbf{b}_i} \quad & \frac{1}{2} \mathbf{b}_i \mathbf{A} \mathbf{b}_i^T + \mathbf{e}_i \mathbf{b}_i^T \\ \text{s.t.} \quad & \mathbf{b}_i \mathbf{1} = 1, \mathbf{b}_i \geq 0, \end{aligned} \quad (15)$$

where $\mathbf{A} = 2\gamma_i \mathbf{I}_n$ and $\mathbf{e}_i = (2\lambda \mathbf{d}_i - \sum_{p=1}^m w_p \mathbf{S}_{p[i,:]})$, and (15) can be simplified as:

$$\begin{aligned} \min_{\mathbf{b}_i} \quad & \frac{1}{2} \|\mathbf{b}_i - \hat{\mathbf{b}}_i\|_2^2 \\ \text{s.t.} \quad & \mathbf{b}_i \mathbf{1} = 1, \mathbf{b}_i \geq 0, \end{aligned} \quad (16)$$

where $\hat{\mathbf{b}}_i = -\frac{\mathbf{e}_i}{2\gamma_i}$. The analytical solution of (16) is as follows [11]:

$$\mathbf{b}_i = \max(\mathbf{b}_i + \beta_i \mathbf{1}_n^T, 0), \quad (17)$$

where β_i can be solved by Newton's method efficiently with the constraint $\mathbf{b}_i \mathbf{1} = 1$.

4.3. Optimization with Respect to \mathbf{F}

When optimizing \mathbf{F} , Equation (11) can be simplified as

$$\begin{aligned} \min_{\mathbf{F}} \quad & \text{tr}(\mathbf{F}^T \mathbf{L} \mathbf{Z} \mathbf{F}) \\ \text{s.t.} \quad & \mathbf{F}^T \mathbf{F} = \mathbf{I}. \end{aligned} \quad (18)$$

According to Ky Fan's theorem, \mathbf{F} is the c smallest eigenvectors before $\mathbf{L} \mathbf{Z}$.

4.4. Initialize the Consensus Matrix \mathbf{Z} and Hyperparameter γ_i

To prefer a sparse discriminative affinity matrix, each row of \mathbf{Z} should have k nonzero values that denote the affinity of each instance corresponding to its initialized neighbors. According to (11), \mathbf{Z} will be naturally sparse by constraining γ_i within reasonable bounds, and by constraining the \mathcal{L}_0 -norm of \mathbf{b}_i to be k with maximal γ_i , the problems is as follows:

$$\max_{\gamma_i} \gamma_i, \quad \text{s.t.} \quad \|\mathbf{b}_i\|_0 = k. \quad (19)$$

Recall the subproblem of optimizing \mathbf{b}_i in (16); its equivalent form can be written as follows:

$$\begin{aligned} \min_{\mathbf{b}_i} \quad & \frac{1}{2} \|\mathbf{b}_i - \frac{\mathbf{e}_i}{2\gamma_i}\|_2^2 \\ \text{s.t.} \quad & \mathbf{b}_i \mathbf{1} = 1, \mathbf{b}_i \geq 0, \end{aligned} \quad (20)$$

where $\mathbf{e}_i = (2\lambda \mathbf{d}_i - \sum_{p=1}^m w_p \mathbf{S}_{p[i,:]})$, and \mathbf{d}_i denotes i -th row of the \mathbf{Z} Euclidean distance matrix squared. We suppose the rows $e_{i1}, e_{i2}, \dots, e_{in}$ are ordered in ascending order. For

each row, $\sum_{p=1}^m w_i \mathbf{S}_{p[i,:]}$ is constant, $\mathbf{d}_{ii} = 0$, and \mathbf{e}_{ii} has the smallest value; therefore, \mathbf{e}_{ii} ranks first. The invalid \mathbf{e}_{ii} should be neglected since the similarity with itself is useless. We use $D_{outN} = \mathbf{e}_{i,k+2}$ to represent the first sample distance that is not in the neighborhood, $D_{sumN} = \sum_{h=2}^{k+1} \mathbf{e}_{ih}$, as the sum of the neighbor distance for the i -th row, and $D_N = \mathbf{e}_{i,j+1}$ indicates the distance between the j -th neighbor in i -th row. To satisfy $\|\mathbf{b}_i\|_0 = k$, the maximal γ_i is as follows [11]:

$$\gamma_i = \frac{k}{2} D_{outN} - \frac{1}{2} D_{sumN}, \quad (21)$$

In the meantime, the initial z_{ij}^* is as follows:

$$z_{ij}^* = \begin{cases} \frac{D_{outN} - D_N}{kD_{outN} - D_{sumN}}, & j \leq k \\ 0, & j > k \end{cases} \quad (22)$$

By initializing a sparse discriminant affinity graph, each row has k nonzero values. Once the initial γ_i has been calculated, γ_i will remain constant throughout the iteration to avoid unnecessary tuning. The initial affinity diagram involves the number of neighbors k , based on the assumption of a balanced cluster; we set $k = \frac{n}{c}$.

4.5. Obtaining \mathbf{G}

Given the consensus matrix \mathbf{Z} , the immediate high-order information can be captured by the diffusion process in Equation (8). The closed-form solution can be computed from Equation (10).

We summarize the overall iterative optimization algorithm of our proposed method in Algorithm 1. Figure 1 demonstrates the process of the whole algorithm.

Algorithm 1 The algorithm of the proposed method.

Input: The m similarity matrix $\mathbf{S}_1, \dots, \mathbf{S}_m$, the self-adaptive parameter $\lambda = 1e^{-4}$, the number c of clusters, the diffusion parameter $\theta = 3$.

Initialize: The neighborhood size $k = \frac{n}{c}$, \mathbf{Z} , and γ_i by Equations (21) and (22).

repeat

 Update \mathbf{w} according to Equation (13);

 Update \mathbf{Z} by solving Equation (17);

 Update \mathbf{F} by solving the smallest eigenvectors;

 Update λ automatically;

until Converges

 Compute graph diffusion consensus matrix \mathbf{G} according to Equation (10);

Output: Perform spectral clustering on consensus matrix \mathbf{G} .

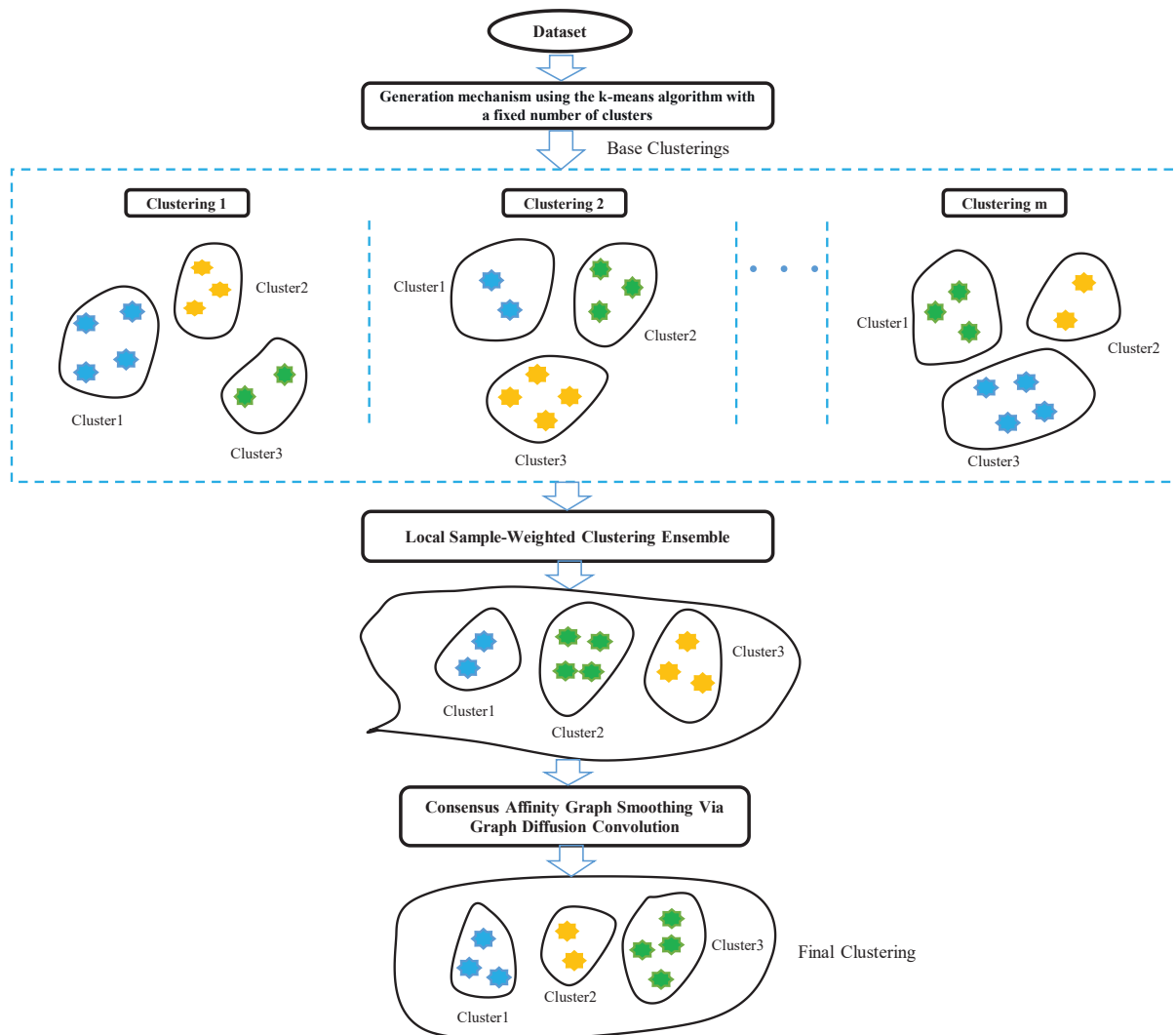


Figure 1. Schematic diagram of the algorithm process.

4.6. Complexity Analysis

Since we need to save and construct the similarity matrix sets \mathcal{S} , the space complexity is $\mathcal{O}(mn^2)$.

The computational complexity of (14) is $\mathcal{O}(n^2m)$. Optimize \mathbf{F} by solving (18) with the eigenvalue decomposition of $\mathbf{L}\mathbf{Z}$, costing $\mathcal{O}(n^2c)$. By computing the closed-formed solution (17) of (16), the computational complexity of (16) is reduced to $\mathcal{O}(nm)$. Optimize \mathbf{Z} to solve n the subproblems (17); the total complexity is $\mathcal{O}(n^2m)$. Therefore, the whole time complexity is $\mathcal{O}((n^2m + n^2c + nm)t)$, where t is the number of iterations.

5. Experiments

This section compares our proposed LSWCEH with the state-of-the-art clustering ensemble methods on benchmark data sets.

5.1. Dataset

In the experiment, this paper used 12 common types of common data sets for the clustering ensemble method, including AR [35], BASEHOCK (<http://qwone.com/jason/20Newsgroups/>, accessed on 6 August 2022), Coil20 [36], CSTR, MSRA25 [37], News4b, ORL400 (<https://jundongli.github.io/scikit-feature/datasets.html>, accessed on 6 August

2022) PCMAC (<http://featureselection.asu.edu/datasets.php>), BBCNews (<http://mlg.ucd.ie/datasets/bbc.html>, accessed on 6 August 2022), Binary Alphadigits (BA) (<http://ida.first.fraunhofer.de/projects/bench/benchmarks.html>, accessed on 6 August 2022), Tr12 [38], and WRP [39]. Different types of data sets can better evaluate the algorithm's performance. The detailed information of the data sets is shown in Table 2.

Table 2. Description of the data sets.

Data Sets	# of Instances	# of Features	# of Classes
AR	840	768	120
BASEHOCK	1993	4862	2
Coil20	1440	1024	20
CSTR	476	1000	4
MSRA25	1799	256	12
News4b	3874	5652	4
ORL400	400	1024	40
PCMAC	1943	3289	4
BA	1404	320	36
BBCNews	737	1000	5
Tr12	313	5804	8
WRP	1560	8460	20

5.2. Compared Algorithms

To demonstrate how the proposed approach can improve the clustering performance, we compared the results of the following algorithms:

- KC [40]: This represents the average of the k-means clustering results. Clustering ensemble methods often use it as the baseline.
- Hypergraph Partitioning Algorithm (HPGA) (the code of the algorithm is provided at <http://www.strehl.com/diss/node80.html>, accessed on 2 October 2022) [4]: A neighborhood partition method based on hypergraphs, the clustering ensemble problem is described as dividing hypergraphs by cutting a minimum number of subgraphs, which uses a given cluster to repartition the data. In order to avoid a large number of scattered partitions, the algorithm sets the hypergraph's hyperedges and the weights of all vertices to equal values.
- Meta Clustering Algorithm (MCLA) (see Section 5.2) [4]: This algorithm transforms the clustering ensemble problem into a cluster consistency problem.
- Dense Representation Clustering Ensemble (DREC) (the code of the algorithm is provided at <https://github.com/zhoujielaoyu/2018-NC-DREC>, accessed on 2 October 2022) [10]: This is an clustering ensemble method based on dense representation. This method introduces a slimming strategy to reduce the input data size by using the information that must be linked between instances to further reduce the time cost of constructing the similarity matrix.
- Probability-Trajectory-based Graph Partitioning (PTGP) (the code of the algorithm is provided at <https://www.researchgate.net/publication/284259332>, accessed on 15 October 2022) [41]: The graph-based method uses sparse graph representation and a random walk process to explore the graph information.
- Locally Weighted Evidence Accumulation (LWEA) (the code of the algorithm is provided at <https://www.researchgate.net/publication/316681928>, accessed on 15 October 2022) [30]: The hierarchical agglomerative clustering ensemble method uses a local weighting strategy.
- Locally Weighted Graph Partitioning (LWGP) (see Section 5.2) [30]: This is a graph partitioning algorithm based on a local weighting strategy. This method constructs a bipartite graph and regards both clusters and objects as graph nodes. In addition, the reliability of clusters is judged by the criterion of entropy.

- Non-negative Matrix Factorization-based Consensus clustering (NMFC) [42]: This is a consensus clustering method based on non-negative matrix factorization.
- Self-Paced Clustering Ensemble (SPCE) [43]: This is a self-paced clustering ensemble method. The method learns from easy to hard and integrates difficulty evaluation and integration learning in a unified framework.
- Multiview Clustering Ensemble (MEC) [44]: This is a robust multi-view clustering ensemble method using low rank and sparse decomposition for the clustering ensemble and noise detection. The method uses low-rank sparse decomposition, explicitly considers the relationship between different views, and detects the noise in each view.
- Robust Spectral Clustering Ensemble (RSEC) [9]: This is a robust clustering ensemble method based on spectral clustering. This method learns the robust representation of the co-association matrix by a low-rank constraint and then finds the consensus partition by spectral clustering.
- LSWCE: This is the proposed method without graph diffusion convolution. To evaluate the effectiveness of graph propagation, we obtain the final clusters from the c connective component in \mathbf{Z} .
- LSWCEH (our code is provided at <https://github.com/sxu-datalab/LSWCEH>, accessed on 2 December 2022): This is the proposed method with high-order graph diffusion based on LSWCE.

5.3. Experimental Settings

K-means results were used as the basic clustering in the experiment. K-means were run 200 times with different initializations on all data sets to obtain 200 basic clustering results, which were further divided into 10 subsets, each with 20 basic results. Finally, we applied all the clustering ensemble methods to each subset and report the average results for the ten subsets. It should be pointed out that there are usually two ways to generate base clusters using the k-means algorithm: (1) the number of clusters c is fixed, which remains unchanged during the running of the algorithm, and c cluster centers are randomly selected each time. (2) Change the c value; allow the c value to change within a certain range; make an appropriate c value interval for each run; then, randomly select c cluster centers. This experiment adopted the first method, which has been widely used.

For our proposed LSWCEH, the self-adaptive hyperparameter λ varies set 1×10^{-4} . Based on the assumption of balanced clusters, the number of neighbors only needs to take the average size of each cluster. The diffusion parameter was set as 3, so the algorithm has fixed parameters and does not need to conduct a grid search.

The clustering performance was evaluated by two widely employed criteria, clustering Accuracy (ACC) and Normalized Mutual Information (NMI). The experimental results were obtained on a desktop with an Intel Core i7 8700K CPU (3.7 GHz), 64-GB RAM, and MATLAB2022a (64-bit).

5.4. Experimental Results

The mean results of the ACC, NMI, and standard deviation are shown in Tables 3 and 4. The bold font in the table indicates the best results, and the next-best results are underlined. According to the experimental results, the following can be seen:

- The results of the LSWCE method were usually better than the compared methods. As seen from the visual display of the clustering accuracy in Table 3, the LSWCE method improved the clustering performance. It can be seen from Table 4 that the clustering result was generally improved under the NMI index. In the LSWCE results, the accuracy evaluation index was improved by 5.4% compared with the second-best algorithm DREC, and the NMI evaluation index was flat compared with the second-best algorithm DREC. The LSWCE method considers the correlation among samples within a local clique. Thus, the redundancy and correlation between these highly similar samples could be primarily alleviated and greatly enhance local discrimination.

- The LSWCEH method achieved the best or second-best results on most data sets. In the LSWCEH results, the accuracy evaluation index was improved by 17.7% compared with the second-best algorithm, DREC, and improved by 15.88% with the NMI evaluation index compared with the second-best algorithm, DREC. Overall, the average values of the ACC and NMI index on all the data sets were better than those of the compared clustering ensemble methods. Such results demonstrated the superiority of our method well.
- The GDC can improve the clustering performance. From the results of LSWCEH, the accuracy evaluation index increased by 11.6% compared with LSWCE, and the NMI evaluation index increased by 17.6%. These results demonstrated the effectiveness of GDC. The clustering results were not significantly improved on a few data sets after GDC. This phenomenon may be because some bad clusters were used as the input, which affects the overall clustering effect.
- The variance of LSWCEH was 2.3% higher than RSEC under the accuracy evaluation index. In Table 4, the variance is 11.9% higher than RSEC under the NMI evaluation index. These results were due to our method being affected by the extreme values. By carefully observing each clustering index of the BASEHOCK and PCMAC data sets obtained by the compared methods, we can see that, although the fluctuation of the variance was small, the clustering effect was unsatisfactory. This means the other compared methods failed on these two data sets and had little effect. LSWCEH greatly improved the two data clustering indexes, which not only increased the variance, but also greatly improved the clustering performance.
- Our approach made the data cluster structure clearer. The process of our method learning the consensus matrix on the ORL data set is illustrated in Figure 2. In Figure 2a is the mean of the input ORL data set. It can be seen from (a) that the similarity matrix in (a) is dense and the cluster structure is unclear. After consensus learning with the local-sample-weighted clustering ensemble, the consensus matrix was sparse. Compared to the original input, as shown in (b), the sample was connected to the neighbor of greater importance, while the neighbor of lesser importance was ignored. The consensus matrix was now noisy, and the local reconstruction using neighborhood relations in all clusters needs to be more accurate. Through higher-order graph diffusion, we obtained a smoother reconstruction consensus matrix. As shown in (c), the graph diffusion strengthened the sample relationship. At this time, the consensus matrix had a weak correlation among some samples due to the graph diffusion, so we need to make it sparse again to obtain a more reliable relationship between samples to obtain a clear structure, as shown in (d). It can also be seen from Figure 3 our method separates the clusters more clearly. Each sample is connected to only a finite number of points.

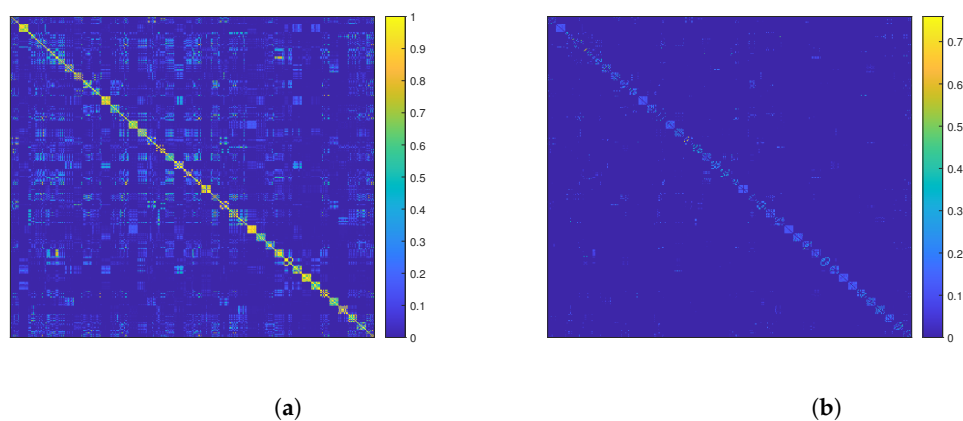


Figure 2. Cont.

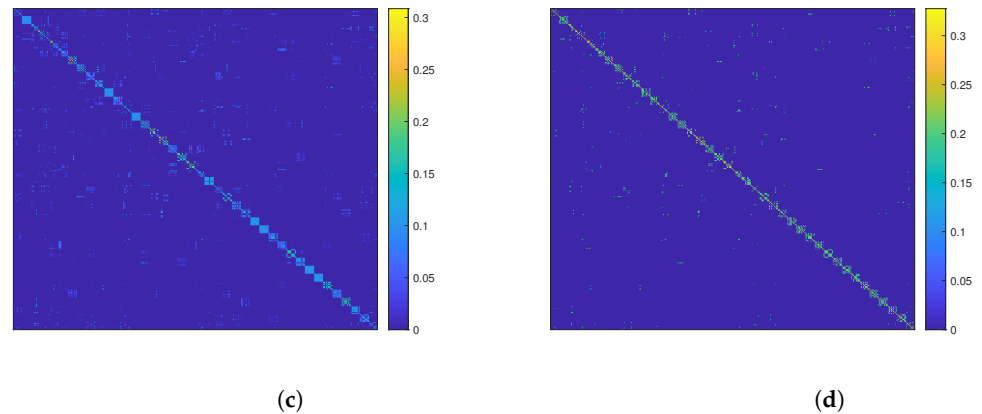


Figure 2. ORL data set clustering structure in the learned consensus matrix at different stages. (a) is the uniform-weight-averaged co-association matrix of the input ORL data set. (b) shows that the sample is connected to the neighbor of greater importance. (c) is the consensus matrix after graph diffusion convolution. (c,d) After making it sparse.

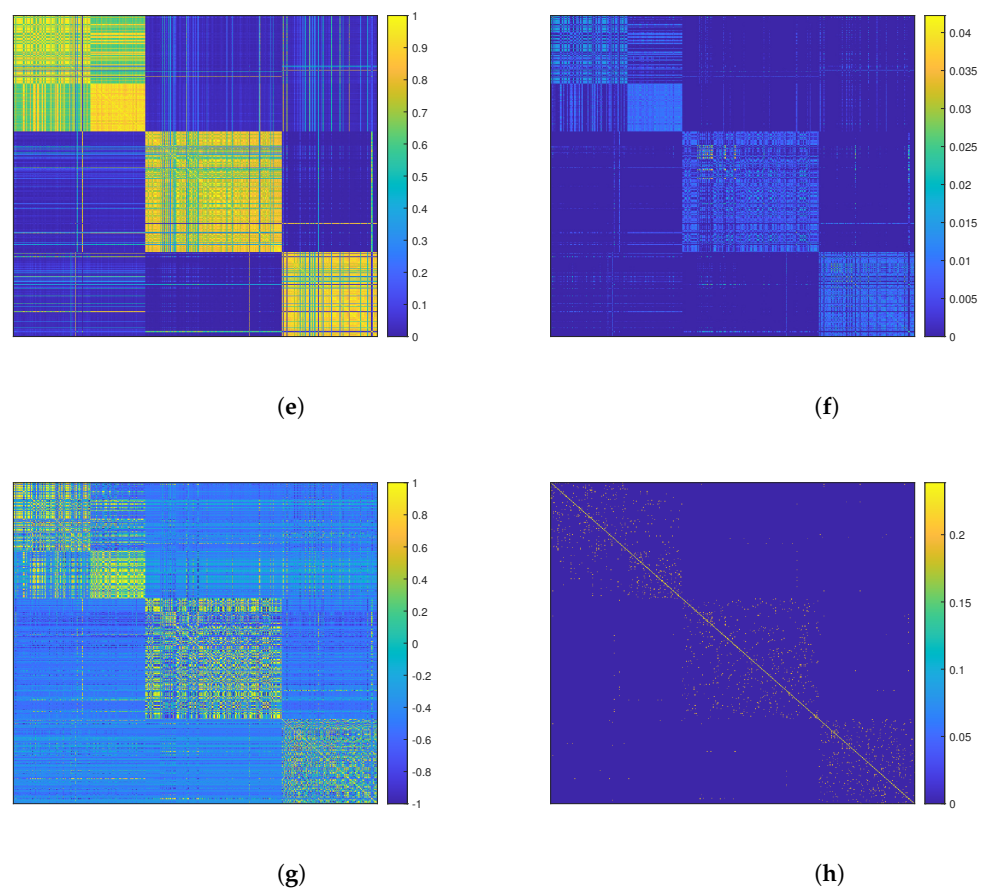


Figure 3. Illustrations of the clustering structure in the learned consensus matrices from different methods on the CSTR data set. (a) is the mean of the input similarity matrix \mathbf{S} . (b,c) Learned consensus matrices by the robust methods MEC and RSEC. (d) is our LSWCEH method. Note that our method of learned consensus is sparse. Each sample is connected to only a finite number of points, and our method separates clusters more clearly.

Table 3. Clustering results measured by the accuracy of the compared methods.

Data Sets	KC	HGPA	MCLA	DREC	PTGP	LWEA	LEGP	MEC	NMFC	RSEC	SPCE	LSWCE	LSWCEH
AR	0.3301 ±0.015	0.3807 ±0.012	0.3337 ±0.115	0.3815 ±0.007	0.3200 ±0.004	0.3898 ±0.013	0.3645 ±0.013	0.2787 ±0.012	0.3692 ±0.006	0.2938 ±0.007	0.3499 ±0.007	<u>0.3955</u> ±0.008	0.4002 ±0.008
BASEHOCK	0.5018 ±0.001	0.5027 ±0.001	0.5029 ±0.001	0.5014 ±0.001	0.5033 ±0.000	0.5009 ±0.001	0.5029 ±0.001	0.5008 ±0.000	0.5029 ±0.001	0.5033 ±0.000	0.5033 ±0.000	<u>0.5057</u> ±0.004	0.7286 ±0.129
Coil20	0.5498 ±0.053	0.5447 ±0.052	0.6695 ±0.023	0.5560 ±0.054	0.4781 ±0.025	0.5817 ±0.025	0.5916 ±0.058	0.5875 ±0.039	0.6615 ±0.030	0.5894 ±0.045	0.6723 ±0.023	<u>0.7021</u> ±0.008	0.7074 ±0.010
CSTR	0.7331 ±0.087	0.2897 ±0.032	0.7966 ±0.029	0.8293 ±0.071	0.8974 ±0.000	0.8019 ±0.004	0.8432 ±0.057	0.9004 ±0.010	0.7842 ±0.084	0.8589 ±0.074	0.8046 ±0.008	0.8949 ±0.009	<u>0.8983</u> ±0.006
MSRA25	0.5094 ±0.048	0.4076 ±0.051	0.5609 ±0.025	0.5522 ±0.036	0.5415 ±0.025	0.5189 ±0.019	0.5364 ±0.048	0.5509 ±0.043	0.5445 ±0.032	0.4818 ±0.035	0.5472 ±0.020	0.5866 ±0.024	<u>0.5619</u> ±0.028
News4b	0.2576 ±0.004	0.2561 ±0.002	0.2608 ±0.003	0.2564 ±0.009	0.2596 ±0.001	0.2573 ±0.001	0.2566 ±0.001	0.2609 ±0.012	0.2580 ±0.009	0.2627 ±0.009	0.2577 ±0.002	<u>0.2651</u> ±0.012	0.5363 ±0.068
ORL400	0.4859 ±0.032	0.5768 ±0.021	0.5878 ±0.012	0.5968 ±0.024	0.6053 ±0.013	0.5735 ±0.029	0.5328 ±0.031	0.4613 ±0.031	0.5745 ±0.016	0.3688 ±0.020	0.5310 ±0.013	<u>0.6205</u> ±0.015	0.6210 ±0.018
PCMAC	0.5050 ±0.000	0.5021 ±0.002	0.5049 ±0.000	0.5049 ±0.000	0.5049 ±0.000	0.5049 ±0.000	0.5049 ±0.000	0.5052 ±0.000	0.5049 ±0.000	0.5049 ±0.000	0.5053 ±0.000	<u>0.5087</u> ±0.008	0.7678 ±0.117
BBCNews	0.5922 ±0.078	0.2246 ±0.005	0.6049 ±0.053	0.6712 ±0.044	0.6608 ±0.000	0.6678 ±0.064	0.6288 ±0.045	0.6780 ±0.050	0.6469 ±0.068	0.5484 ±0.030	0.6362 ±0.060	0.6967 ±0.039	<u>0.6863</u> ±0.044
BA	0.4088 ±0.019	0.3620 ±0.018	0.4542 ±0.009	0.4724 ±0.009	0.4717 ±0.008	0.4033 ±0.010	0.4190 ±0.029	0.4269 ±0.022	0.4463 ±0.020	0.2958 ±0.069	0.3627 ±0.012	<u>0.4729</u> ±0.005	0.4778 ±0.009
Tr12	0.4739 ±0.078	0.4920 ±0.060	0.5466 ±0.085	0.6198 ±0.048	0.5827 ±0.007	0.6240 ±0.068	0.5623 ±0.081	0.5958 ±0.045	0.5898 ±0.055	0.4048 ±0.112	0.4537 ±0.070	<u>0.6348</u> ±0.055	0.6364 ±0.031
WRP	0.4202 ±0.047	0.3610 ±0.021	0.4876 ±0.061	0.5144 ±0.042	0.5392 ±0.014	0.5785 ±0.013	0.4799 ±0.040	0.4622 ±0.046	0.4952 ±0.041	0.4893 ±0.099	0.4376 ±0.038	0.5228 ±0.033	0.5810 ±0.052
Avg	0.4806 ±0.039	0.4083 ±0.023	0.5259 ±0.035	0.5380 ±0.029	0.5320 ±0.008	0.5335 ±0.021	0.5186 ±0.034	0.5174 ±0.026	0.5315 ±0.030	0.4668 ±0.042	0.5051 ±0.021	<u>0.5673</u> ±0.018	0.6335 ±0.043

Table 4. Clustering results measured by the NMI of the compared methods.

Data Sets	KC	HGPA	MCLA	DREC	PTGP	LWEA	LEGP	MEC	NMFC	RSEC	SPCE	LSWCE	LSWCEH
AR	0.6533 ± 0.0093	0.7039 ± 0.0065	0.6878 ± 0.0047	0.6911 ± 0.0072	0.6327 ± 0.0058	0.6748 ± 0.0073	0.6825 ± 0.0086	0.5634 ± 0.0154	0.6948 ± 0.0037	0.5828 ± 0.0147	0.7297 ± 0.0021	0.7083 ± 0.0048	<u>0.7108</u> ± 0.0038
BASEHOCK	0.0024 ± 0.0018	0.0000 ± 0.0000	0.0041 ± 0.0010	0.0017 ± 0.0011	0.0045 ± 0.0000	0.0008 ± 0.0010	0.0041 ± 0.0010	0.0005 ± 0.0000	0.0040 ± 0.0013	0.0045 ± 0.0000	<u>0.0045</u> ± 0.0000	0.0006 ± 0.0014	0.2924 ± 0.2209
Coil20	0.7061 ± 0.0271	0.6756 ± 0.0224	0.7570 ± 0.0121	0.7300 ± 0.0542	0.6454 ± 0.0183	0.7317 ± 0.0152	0.7289 ± 0.0302	0.7363 ± 0.0221	0.7635 ± 0.0110	0.7360 ± 0.0181	0.7810 ± 0.0275	<u>0.7950</u> ± 0.0092	0.7959 ± 0.0062
CSTR	0.6390 ± 0.0635	0.0150 ± 0.0164	0.6734 ± 0.0190	0.7100 ± 0.0713	0.7829 ± 0.0000	0.6902 ± 0.0080	0.7183 ± 0.0431	<u>0.7726</u> ± 0.0171	0.6944 ± 0.0251	0.7526 ± 0.0439	0.6703 ± 0.0183	0.7507 ± 0.0193	0.7578 ± 0.0135
MSRA25	0.5820 ± 0.0439	0.4481 ± 0.0729	0.6098 ± 0.0185	0.6364 ± 0.0358	0.6364 ± 0.0168	0.6250 ± 0.0170	0.5983 ± 0.0349	0.6182 ± 0.0306	0.6165 ± 0.0221	0.5496 ± 0.0321	<u>0.6477</u> ± 0.0084	0.6651 ± 0.0075	0.6281 ± 0.0282
News4b	0.0063 ± 0.0030	0.0002 ± 0.0002	0.0030 ± 0.0017	0.0064 ± 0.0088	0.0097 ± 0.0007	0.0042 ± 0.0006	0.0074 ± 0.0020	0.0075 ± 0.0065	0.0071 ± 0.0043	0.0104 ± 0.0058	<u>0.0114</u> ± 0.0018	0.0090 ± 0.0059	0.3292 ± 0.0516
ORL400	0.6898 ± 0.0195	0.7616 ± 0.0079	0.7534 ± 0.0057	0.7741 ± 0.0237	0.7688 ± 0.0038	0.7616 ± 0.0097	0.7270 ± 0.0164	0.6477 ± 0.0205	0.7547 ± 0.0086	0.5802 ± 0.0206	0.7663 ± 0.0047	<u>0.7795</u> ± 0.0106	0.7821 ± 0.0079
PCMAC	0.0001 ± 0.0000	0.0000 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0000 ± 0.0000	0.0001 ± 0.0000	0.0001 ± 0.0000	0.0000 ± 0.0000	<u>0.0006</u> ± 0.0009	0.3329 ± 0.1627
BBCNews	0.4133 ± 0.0776	0.0050 ± 0.0041	0.4246 ± 0.0399	<u>0.5209</u> ± 0.0440	0.5123 ± 0.0000	0.4817 ± 0.0551	0.4268 ± 0.0484	0.5233 ± 0.0456	0.4548 ± 0.0538	0.3977 ± 0.0295	0.4927 ± 0.0247	0.5077 ± 0.0346	0.5034 ± 0.0291
BA	0.5700 ± 0.0101	0.5296 ± 0.0082	0.5866 ± 0.0045	0.6033 ± 0.0090	<u>0.6041</u> ± 0.0062	0.5540 ± 0.0057	0.5710 ± 0.0185	0.5693 ± 0.0198	0.5856 ± 0.0071	0.4078 ± 0.0932	0.5953 ± 0.0029	0.6038 ± 0.0041	0.6044 ± 0.0044
Tr12	0.3859 ± 0.0805	0.4105 ± 0.0585	0.4759 ± 0.0949	0.6028 ± 0.0476	0.6177 ± 0.0036	0.5429 ± 0.0535	0.5374 ± 0.0515	0.5768 ± 0.0525	0.5501 ± 0.0547	0.2321 ± 0.1705	0.2752 ± 0.0787	0.6053 ± 0.0395	<u>0.6063</u> ± 0.0379
WRP	0.5180 ± 0.0237	0.4584 ± 0.0150	0.5684 ± 0.0239	0.5994 ± 0.0422	<u>0.6025</u> ± 0.0039	0.5480 ± 0.0150	0.5561 ± 0.0228	0.5558 ± 0.0181	0.5723 ± 0.0227	0.5026 ± 0.0935	0.3556 ± 0.0586	0.5875 ± 0.0113	0.6103 ± 0.0184
Avg	0.4305 ± 0.0300	0.3340 ± 0.0177	0.4620 ± 0.0188	<u>0.4897</u> ± 0.0288	0.4848 ± 0.0049	0.4679 ± 0.0157	0.4631 ± 0.0231	0.4643 ± 0.0207	0.4748 ± 0.0179	0.3964 ± 0.0435	0.4441 ± 0.0190	0.4823 ± 0.0124	0.5675 ± 0.0487

6. Conclusions

In this article, we proposed a novel LSWCEH method. Unlike traditional clustering ensemble methods that use the number of all sample neighbors in integration learning,

we distinguished the order of sample importance in our learning. In the LSWCEH framework, we used an efficient graph diffusion algorithm to improve the clustering result. We conducted extensive experiments on the benchmark data set, and the experimental results showed that our method had significant performance improvement over the base clustering results.

Author Contributions: Conceptualization, J.G. and Y.L.; methodology, J.G.; software, J.G.; validation, J.G. and Y.L.; investigation, Y.L.; resources, J.G.; data curation, L.D.; writing—original draft preparation, J.G.; writing—review and editing, J.G.; visualization, Y.L.; supervision, Y.L.; project administration; funding acquisition, L.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Natural Science Foundation of China (61976129).

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank the Reviewers for providing valuable comments and suggestions.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three letter acronym
LD	Linear dichroism

References

- Topchy, A.; Jain, A.K.; Punch, W. A mixture model for clustering ensembles. In Proceedings of the 2004 SIAM International Conference on Data Mining, Lake Buena Vista, FL, USA, 22–24 April 2004; pp. 379–390.
- Zhang, M. Weighted clustering ensemble: A review. *Pattern Recognit.* **2022**, *124*, 108428.
- Zhou, P.; Wang, X.; Du, L.; Li, X. clustering ensemble via structured hypergraph learning. *Inf. Fusion* **2022**, *78*, 171–179.
- Strehl, A.; Ghosh, J. clustering ensembles—A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **2002**, *3*, 583–617.
- Li, T.; Rezaeipana, A.; El Din, E.M.T. An ensemble agglomerative hierarchical clustering algorithm based on clusters clustering technique and the novel similarity measurement. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 3828–3842.
- Fred, A.L.; Jain, A.K. Combining multiple clusterings using evidence accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 835–850.
- Li, T.; Ding, C.; Jordan, M.I. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, USA, 28–31 October 2007; pp. 577–582.
- Tao, Z.; Liu, H.; Fu, Y. Simultaneous clustering and ensemble. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 1546–1552.
- Tao, Z.; Liu, H.; Li, S.; Fu, Y. Robust spectral clustering ensemble. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 24–28 October 2016; pp. 367–376.
- Zhou, J.; Zheng, H.; Pan, L. Clustering ensemble based on dense representation. *Neurocomputing* **2019**, *357*, 66–76.
- Li, L.; Wang, S.; Liu, X.; Zhu, E.; Shen, L.; Li, K.; Li, K. Local sample-weighted multiple kernel clustering with consensus discriminative graph. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–14. <http://doi.org/10.1109/TNNLS.2022.3184970>.
- Tang, C.; Liu, X.; Zhu, X.; Xiong, J.; Li, M.; Xia, J.; Wang, X.; Wang, L. Feature selective projection with low-rank embedding and dual Laplacian regularization. *IEEE Trans. Knowl. Data Eng.* **2019**, *32*, 1747–1760.
- Nie, F.; Wang, X.; Huang, H. Clustering and projected clustering with adaptive neighbors. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 977–986.
- Bai, S.; Zhou, Z.; Wang, J.; Bai, X.; Jan Latecki, L.; Tian, Q. Ensemble diffusion for retrieval. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 774–783.
- Klicpera, J.; Weißenberger, S.; Günnemann, S. Diffusion improves graph learning. *arXiv* **2019**, arXiv:1911.05485.

16. Zhou, Z.H.; Tang, W. Clusterer ensemble. *Knowl.-Based Syst.* **2006**, *19*, 77–83.
17. Li, F.; Qian, Y.; Wang, J.; Liang, J. Multigranulation information fusion: A Dempster-Shafer evidence theory-based clustering ensemble method. *Inf. Sci.* **2017**, *378*, 389–409.
18. Iam-On, N.; Boongoen, T.; Garrett, S. LCE: A link-based clustering ensemble method for improved gene expression data analysis. *Bioinformatics* **2010**, *26*, 1513–1519.
19. Iam-On, N.; Boongeon, T.; Garrett, S.; Price, C. A link-based clustering ensemble approach for categorical data clustering. *IEEE Trans. Knowl. Data Eng.* **2010**, *24*, 413–425.
20. Liu, H.; Zhao, R.; Fang, H.; Cheng, F.; Fu, Y.; Liu, Y.Y. Entropy-based consensus clustering for patient stratification. *Bioinformatics* **2017**, *33*, 2691–2698.
21. Fern, X.Z.; Brodley, C.E. Solving clustering ensemble problems by bipartite graph partitioning. In Proceedings of the Twenty-First International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004; p. 36.
22. Huang, D.; Lai, J.H.; Wang, C.D. Combining multiple clusterings via crowd agreement estimation and multi-granularity link analysis. *Neurocomputing* **2015**, *170*, 240–250.
23. Yu, Z.; Wong, H.S.; Wang, H. Graph-based consensus clustering for class discovery from gene expression data. *Bioinformatics* **2007**, *23*, 2888–2896.
24. Jia, J.; Xiao, X.; Liu, B.; Jiao, L. Bagging-based spectral clustering ensemble selection. *Pattern Recognit. Lett.* **2011**, *32*, 1456–1467.
25. Liu, H.; Liu, T.; Wu, J.; Tao, D.; Fu, Y. Spectral clustering ensemble. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 715–724.
26. Mimaroglu, S.; Aksehirli, E. DICLNS: Divisive clustering ensemble with automatic cluster number. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2011**, *9*, 408–420.
27. Zhong, C.; Yue, X.; Lei, J. Visual hierarchical cluster structure: A refined co-association matrix based visual assessment of cluster tendency. *Pattern Recognit. Lett.* **2015**, *59*, 48–55.
28. Hu, J.; Li, T.; Wang, H.; Fujita, H. Hierarchical clustering ensemble model based on knowledge granulation. *Knowl.-Based Syst.* **2016**, *91*, 179–188.
29. Wang, T. CA-Tree: A hierarchical structure for efficient and scalable coassociation-based clustering ensembles. *IEEE Trans. Syst. Man, Cybern. Part B (Cybern.)* **2010**, *41*, 686–698.
30. Huang, D.; Wang, C.D.; Lai, J.H. Locally weighted clustering ensemble. *IEEE Trans. Cybern.* **2017**, *48*, 1460–1473.
31. Lin, Z.; Kang, Z.; Zhang, L.; Tian, L. Multi-view Attributed Graph Clustering. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 1872–1880.
32. Kang, Z.; Lin, Z.; Zhu, X.; Xu, W. Structured graph learning for scalable subspace clustering: From single view to multiview. *IEEE Trans. Cybern.* **2022**, *52*, 8976–8986.
33. Chung, F. The heat kernel as the pagerank of a graph. *Proc. Natl. Acad. Sci. USA* **2007**, *104*, 19735–19740.
34. Fan, K. On a theorem of Weyl concerning eigenvalues of linear transformations: II. *Proc. Natl. Acad. Sci. USA* **1950**, *36*, 31–35.
35. Wang, H.; Nie, F.; Huang, H. Globally and locally consistent unsupervised projection. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; Volume 28.
36. Cai, D.; He, X.; Han, J.; Huang, T.S. Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 1548–1560.
37. Winn, J.; Jojic, N. Locus: Learning object classes with unsupervised segmentation. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV’05), Montreal, QC, Canada, 10–17 October 2021; Volume 1, pp. 756–763.
38. Zhao, Y.; Karypis, G. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Mach. Learn.* **2004**, *55*, 311–331.
39. Maoz, Z.; Henderson, E.A. The world religion data set, 1945–2010: Logic, estimates, and trends. *Int. Interact.* **2013**, *39*, 265–291.
40. MacQueen, J. Classification and analysis of multivariate observations. In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967; pp. 281–297.
41. Huang, D.; Lai, J.H.; Wang, C.D. Robust clustering ensemble using probability trajectories. *IEEE Trans. Knowl. Data Eng.* **2015**, *28*, 1312–1326.
42. Li, T.; Ding, C. Weighted consensus clustering. In Proceedings of the 2008 SIAM International Conference on Data Mining, Alexandria, VA, USA, 28–30 April 2008; pp. 798–809.
43. Zhou, P.; Du, L.; Liu, X.; Shen, Y.D.; Fan, M.; Li, X. Self-paced clustering ensemble. *IEEE Trans. Neural Networks Learn. Syst.* **2020**, *32*, 1497–1511.
44. Tao, Z.; Liu, H.; Li, S.; Ding, Z.; Fu, Y. From clustering ensemble to multi-view clustering. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 2843–2849.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.