International Journal of Innovative
Computing, Information and Control
Volume 17, Number 4, August 2021

ICIC International ©2021 ISSN 1349-4198
pp. 1327–1343

SOFT SUBSPACE CLUSTERING ENSEMBLE BASED ON HEDONIC GAMES

Man Li and Lihong Wang*

School of Computer and Control Engineering
Yantai University
No. 30, Qingquan Road, Laishan District, Yantai 264005, P. R. China
Lman0720@163.com; *Corresponding author: wanglh@ytu.edu.cn

Received December 2020; revised April 2021

Abstract. *Clustering ensemble aims at producing consensual, better-quality partitions by integrating base partitions. Hedonic Game based Clustering Ensemble (HGCE) exploits a hedonic game to discover the best coalition structure for all data points. However, the consensus partition obtained by HGCE has much more clusters than the true classes available when the game reaches a Nash equilibrium. Additionally, HGCE evaluates the pairwise similarity by co-association matrix without considering the qualities of base partitions. In this paper, Soft Subspace clustering Ensemble based on Hedonic games (SSEH) is proposed to integrate a set of base partitions generated by a soft subspace clustering algorithm ERKM (Entropy Regularization K-Means). The negative entropy coefficient of ERKM helps to generate diversified base partitions, and a new pairwise similarity is evaluated on the basis of the cluster stability. Moreover, the clusters endowed with Nash equilibrium are merged step by step to the ground truth number of classes by minimizing the loss of the social welfare. The experiment results evaluated by three metrics show that two versions of the proposed SSEH have different advantages in terms of average ranks and W/T/L (Wins/Ties/Losses) on 20 test datasets.*

Keywords: Soft subspace clustering, Clustering ensemble, Hedonic game, Nash equilibrium, Cluster stability

1. Introduction.

As a classical data analysis method, clustering has been widely used in computer, biology, economics and other fields [1]. For high-dimensional data, clusters usually exist in subspaces of the entire data space. Maybe the values of data on some dimensions are uniformly distributed, and clusters are irrelevant to these features (attributes). Given two neighboring points in the cluster, they can be far apart on the irrelevant attributes. If all attributes participate in clustering, irrelevant attributes will deteriorate the clustering performance. Therefore, the subspace clustering is proposed to find different subspaces and the clusters in subspaces.

Subspace clustering can be divided into hard subspace clustering and soft subspace clustering (SSC) [2]. The goal of hard subspace clustering is to identify accurate subspaces [3, 4], while soft subspace clustering assigns a weight value to each dimension to measure its contribution to a cluster. Soft subspace clustering is generally considered as an extension of traditional feature-weighted clustering [5, 6]. This study focuses on the negative entropy-based soft subspace clustering. In 2004, Friedman and Meulman [7] introduced the negative entropy to the objective function for the first time to stimulate more dimensions to contribute to the identification of clusters in subspace clustering. Since then, many negative entropy-based SSC algorithms have been proposed successively. On the basis of K-means [8, 9, 10] or FCM [11, 12], the negative entropy term was included

DOI: 10.24507/ijicic.17.04.1327

as a part of the objective functions. Experiments show that, the clustering results are sensitive to the negative entropy coefficient, and the large range of the negative entropy coefficient makes it hard to accurately specify the best value [8, 10]. This motivates us to exploit ensemble techniques to combine multiple base partitions generated by negative entropy-based SSC algorithms to obtain a better-quality and more robust consensus partition for a dataset.

There are two key issues in clustering ensemble, one is the generation of diversified base partitions, and the other is the design of the consensus function to combine base partitions. Firstly, diversified base partitions can be obtained in two ways. 1) Different clustering algorithms are applied to finding the cluster structure from multiple views. For example, Yoon et al. [13] used K-means, hierarchical algorithm and principal component analysis-based algorithm to generate base partitions. 2) The same clustering algorithms are used with different parameters or initial values. K-means [14, 15, 16, 17, 18, 19] is usually adopted as the base partition generator due to its sensitivity to the initial value. Secondly, the design of consensus function is usually based on the co-association matrix [18], binary matrix [16] or hypergraph [20]. The co-association matrix records the frequency of data point pairs in the same cluster, which is usually used to measure the similarity between data points. The binary matrix focuses on describing the relationship between data points and base clusters, and the hypergraph is used to measure the membership of data points to the clusters.

Moreover, the game theory was investigated for the purpose of clustering ensemble in 2011 [21]. Garg et al. [22] mapped cluster formation to coalition formation in cooperative games, and used the Shapely value of the patterns to identify clusters and cluster representatives. Feldman et al. [23] dealt with the clustering problem based on the hedonic game model, and Sandes and André [14] conducted an in-depth analysis of the hedonic game theory and proposed Hedonic Game based Clustering Ensemble (HGCE) to discover the best coalition structure for all data points. However, the consensus partition obtained by HGCE has much more clusters than the true classes available when the game reaches a Nash equilibrium [14]. Additionally, HGCE evaluates the pairwise similarity by the co-association matrix without considering the qualities of base partitions.

In this study, Soft Subspace clustering Ensemble based on Hedonic games (SSEH) is proposed to integrate a set of base partitions generated by a soft subspace clustering algorithm ERKM (Entropy Regularization K-Means) [10]. We replace the K-means by ERKM as the base partition generator for its high sensitivity to the negative entropy coefficient, and expect that diversified base partitions can be generated. Since the consensus partition obtained by HGCE has much more clusters than the true classes available, it is necessary to find a way to merge the small clusters to reach the real ones. In the SSEH, a new pairwise similarity measurement based on the cluster stability is proposed to improve the co-association matrix for the consensus partition, and merging is conducted for a reasonable cluster number by minimizing the loss of social welfare. Additionally, in the ensemble of subspace clustering, it is inevitable to answer the question, "What is the subspace of the consensus partition"? We propose a convex optimization-based method to retrieve the subspace of the consensus partition in the view of feature weights. Finally, potential applications of SSEH are discussed.

The rest of this paper is organized as follows. In Section 2, we review the related work of soft subspace clustering with negative entropy and HGCE. In Section 3, we propose our algorithm. The experimental results are introduced in Section 4, and finally Section 5 summarizes the paper.

2. **Related Work.**

2.1. **Negative entropy-based soft subspace clustering.**

The Entropy Regularization K-Means (ERKM) algorithm was proposed in 2019, in which all subspaces share a feature weighting vector $W = \{w_1, w_2, \dots, w_D\}$. Its objective function is defined as follows [10]:

$$\begin{aligned}\min J_{\text{ERKM}}(W,U,V) = &\sum_{p=1}^{K}\sum_{i=1}^{N} u_{pi}\sum_{j=1}^{D} w_j(x_{ij}-v_{pj})^2 + \gamma\sum_{j=1}^{D} w_j\log w_j \\ &-\eta\sum_{p=1}^{K}\sum_{i=1}^{N}(1-u_{pi})\sum_{j=1}^{D} w_j(x_{ij}-v_{pj})^2\end{aligned}$$

$$\text{s.t.}\quad \begin{cases}\displaystyle\sum_{j=1}^{D} w_j = 1, \quad 0 < w_j < 1 \\ \displaystyle\sum_{p=1}^{K} u_{pi} = 1, \quad u_{pi} \in \{0,1\}\end{cases} \tag{1}$$

where $X = (x_{ij})_{N\times D}$ is the dataset of N objects with D attributes, $U = (u_{pi})_{K\times N}$ is the membership matrix, and $V = (v_{pj})_{K\times D}$ is the matrix of centers, with each row a center. The first term of the objective function is the sum of weighted within-cluster distance, the second one is the negative entropy term, and the third part is the sum of weighted between-cluster distance. $\eta > 0$ is a trade-off parameter that balances the within-cluster distance and the between-cluster distance.

Despite its outstanding performance, ERKM needs to determine the coefficient γ in advance and the clustering results are sensitive to γ, which limits the applications of ERKM. Therefore, we exploit the ensemble technique to integrate the base partitions generated by ERKM and get a better-quality clustering result.

2.2. **HGCE.**

Let $X = \{x_1, x_2, \dots, x_N\}$ denote a dataset with N objects, and CS be a base partition of X with K clusters generated by a clustering algorithm, such that $CS = \{C_k | k = 1, 2, \dots, K\}$, $\cup_{k=1}^{K} C_k = X$, and $C_k \cap C_{k'} = \emptyset$, $\forall k \neq k'$. Assuming $\Pi = \left\{CS^1, CS^2, \dots, CS^M\right\}$ is a set of M base partitions of X, the clustering ensemble aims at combining multiple base partitions to obtain a better-quality and more robust consensus partition.

In the cooperative game model [21], how to form a coalition and distribute the payoff among individuals in the coalition are two key problems. The hedonic game [23] is a simplified cooperative game, which only considers the formation of coalitions. In the hedonic game scenario, players have their own preferences for the coalition and decide whether or not to form the same coalition based on their relationship with other players.

Sandes and André [14] regarded data points as players and clusters as coalitions, and then a base partition $CS = \{C_k | k = 1, 2, \dots, K\}$ is a coalition structure. Assume v_i to be the preference function of player x_i, and let v_{ij} denote the utility value of player x_j for player x_i, then $v_{ij} = v_i(j) = v_j(i) = v_{ji}$ by symmetry, and the preference of player x_i for a coalition C_k is $v_i(C_k) = \sum_{j\in C_k} v_{ij}$.

Given a coalition structure CS, the social welfare of all players is the sum of the utilities of all the players in the coalition structure, i.e.,

$$v(CS) = \sum_{\{i,j\}:\exists C_k\in CS, \{i,j\}\subset C_k} v_{ij}. \tag{2}$$

If the player x_i moves from its current coalition CS_i to another coalition C_k, then the social welfare difference between the new coalition structure CS' with CS is

$$v\left(CS'\right) - v(CS) = v_i(C_k) - v_i(CS_i). \tag{3}$$

The coalition structure CS' will have a higher social welfare than that of CS if $v_i(C_k) > v_i(CS_i)$. Thus, the hedonic game reaches a Nash equilibrium when each player's preference for the current coalition reaches its maximum value and does not have to move to any other coalition, i.e., the current coalition structure is optimal and Nash stable for all players [14].

Sandes and André [14] proposed the HGCE by defining the player x_i's preference for coalition C_k as

$$v_i(C_k) = \sum_{j \in C_k} v_{ij} = \sum_{j \in C_k} sim(x_i, x_j), \tag{4}$$

$$sim(x_i, x_j) = \left|\left\{CS^m | CS_i^m = CS_j^m\right\}\right| - \left|\left\{CS^m | CS_i^m \neq CS_j^m\right\}\right| \tag{5}$$

where $sim(x_i,\, x_j)$ is the value of player x_j for player x_i, $v_{ij} = sim(x_i,\, x_j)$, and $\Pi = \left\{CS^1, CS^2, \dots, CS^M\right\}$ is a set of M base partitions of X, $x_i \in CS_i^m$, $x_j \in CS_j^m$, $m = 1, 2, \dots, M$.

HGCE creates the co-association matrix by treating all base partitions in the same way. However, the clusters in base partitions are different in qualities, since a cluster may be stable or unstable [17, 24, 25]. We consider the cluster stability into hedonic games to improve HGCE further.

2.3. Ensemble-driven Cluster Index (ECI).

Huang et al. [17] considered the diversity and stability of base partitions and proposed locally weighted ensemble clustering, in which the Ensemble-driven Cluster Index (ECI) was defined to measure the stability of cluster C_i with respect to the M base partitions.

$$ECI(C_i) = e^{-\frac{H^{\Pi}(C_i)}{\theta \cdot M}} \tag{6}$$

where $\Pi = \left\{CS^1, CS^2, \dots, CS^M\right\}$ is a set of M base partitions of X, $\theta > 0$ is a parameter to adjust the influence of the cluster uncertainty over ECI, and it is suggested that the parameter θ should be set in the interval of $[0.2, 1]$ [17]. Moreover, $H^{\Pi}(C_i)$ is the uncertainty of cluster C_i with respect to the entire ensemble Π, which equals the sum of $H^m(C_i)$, i.e.,

$$H^{\Pi}(C_i) = \sum_{m=1}^{M} H^m(C_i) \tag{7}$$

and

$$H^m(C_i) = -\sum_{j=1}^{N_m} \frac{\left|C_i \cap C_j^m\right|}{|C_i|} \log \frac{\left|C_i \cap C_j^m\right|}{|C_i|} \tag{8}$$

where $H^m(C_i)$ is the uncertainty of cluster C_i with respect to the base partition CS^m in the ensemble Π, C_j^m denotes the jth cluster of the CS^m, and N_m is the number of clusters in the CS^m.

We apply this index to the similarity measure of hedonic games and expect to get stable clusters.

3. The Proposed SSEH.

3.1. A new pairwise similarity based on cluster stability.

For a stable cluster C_i, the objects in C_i usually belong to the same cluster of all base partitions. Huang et al. showed that, a stable cluster has a greater ECI value [17]. Thus, we combine ECI with the co-association matrix to measure the similarity between data points, and define a new pairwise similarity based on the cluster stability index as follows

$$\begin{aligned} sim_{new}(x_i, x_j) = {} & \frac{n_1}{M} \sum_{CS_i^m = CS_j^m} ECI(CS_i^m) \\ & - \frac{(M - n_1)}{M} \sum_{CS_i^m \neq CS_j^m} \max\left(ECI\left(CS_i^m\right), ECI\left(CS_j^m\right)\right) \end{aligned} \tag{9}$$

where n_1 is the number of times the data points x_i and x_j are clustered together in the base partitions of Π, $x_i \in CS_i^m$, $x_j \in CS_j^m$ and ECI is used to weigh clusters. Specially, $sim_{new}(x_i, x_i) = 0$.

Obviously, $sim_{new}(x_i, x_j)$ is symmetry, $sim_{new}(x_i, x_j) = sim_{new}(x_j, x_i)$. $sim_{new}(x_i, x_j)$ is also the value of player x_j for player x_i, $v_{ij} = sim_{new}(x_i, x_j)$.

The first term represents the average cluster stability weighed by the times that x_i and x_j are clustered together in the base partitions of Π. If x_i and x_j are not clustered together in a base partition, we select the larger one of $ECI\left(CS_i^m\right)$ and $ECI\left(CS_j^m\right)$ to reduce the similarity between x_i and x_j. In summary, the more times two data points clustered together, the more stable the cluster, and the larger the pairwise similarity.

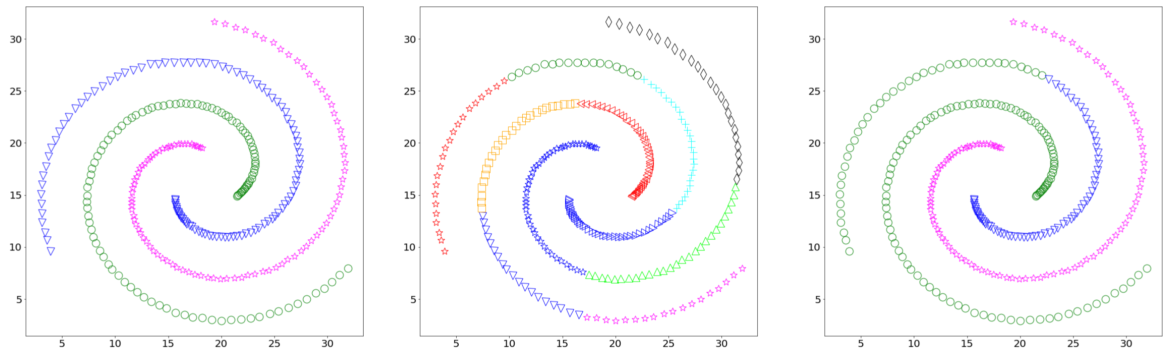
Furthermore, we redefine the preference of player x_i over a coalition (cluster) C_k as

$$v_i(C_k) = \sum_{j \in C_k} sim_{new}(x_i, x_j). \tag{10}$$

And the social welfare of a coalition structure $CS = \{C_k | k = 1, \dots, K\}$, $v(CS)$, is defined as Equation (2).

Following HGCE, $v(CS)$ can be deemed as an exact potential function with symmetric preferences, and any coalition structure maximizing $v(CS)$ is Nash stable [14, 26]. Therefore, the hedonic game with the new pairwise similarity will find at least one coalition structure endowed with Nash stability.

3.2. **Merging clusters.** Generally, the cluster number is much higher than the true number of clusters when the HGCE reaches a Nash equilibrium [14], as shown in Figures 1(a) and 1(b). Figure 1 shows that three long curly clusters in the Spiral dataset [18] are broken into 11 clusters by HGCE. Thus, it is necessary to recover the true clusters by merging the small clusters. We merge the small clusters by minimizing the loss of social welfare until the cluster number equals the ground truth number of clusters.



(a) Ground truth clusters (b) 11 clusters by HGCE (c) 3 clusters after merging

FIGURE 1. The clustering results of the Spiral dataset

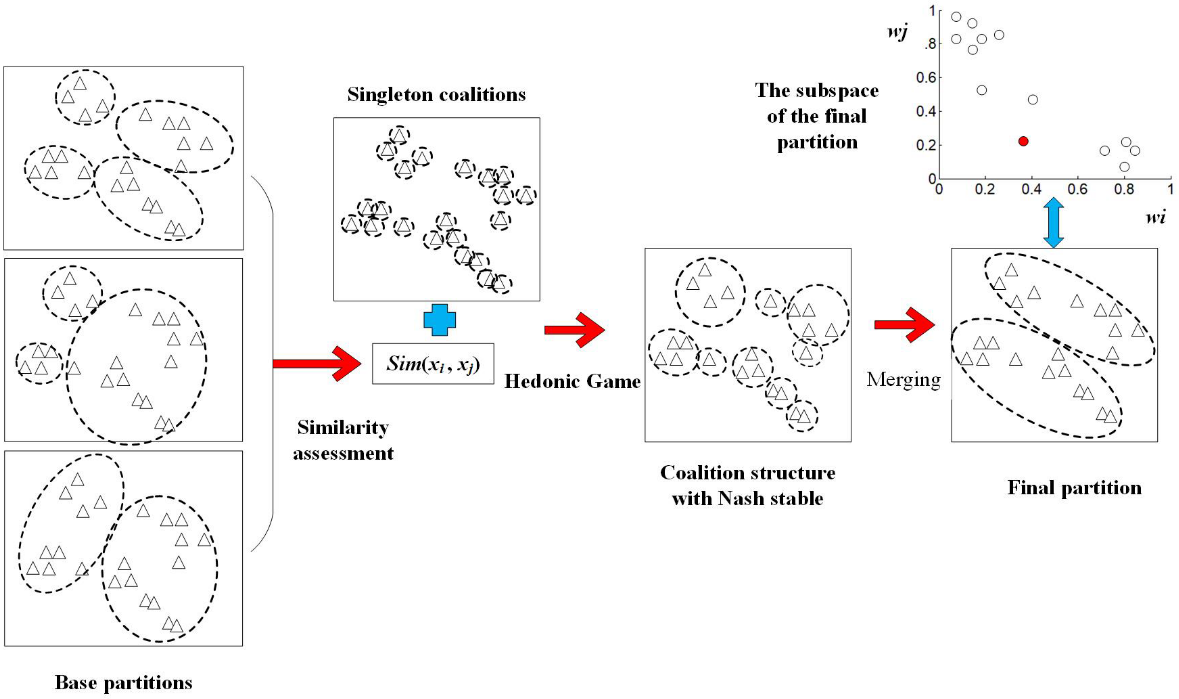


Figure 2. The framework of SSEH

If two clusters C_i and C_j in the coalition structure CS are merged, the new social welfare of CS' will be updated as follows

$$v(CS') = v(CS) + \sum_{x_i \in C_i, x_j \in C_j} v_{ij}. \tag{11}$$

We merge the small clusters by minimizing the loss of social welfare, so we select two clusters C_i and C_j such that

$$\arg\min_{C_i, C_j} \sum_{x_i \in C_i, x_j \in C_j} v_{ij}. \tag{12}$$

Figure 1(c) shows the clusters after merging, two clusters are recovered correctly, and the third one is recovered partly and merged with another long cluster.

The framework of the proposed SSEH is shown in Figure 2. Multiple base partitions are generated by the algorithm ERKM, and then ECI is used to measure the similarity between data points, and the hedonic game is exploited to obtain the clusters with Nash stability. Finally, the clusters are merged to the true number of classes available, and the subspace of the final partition is retrieved.

The details of the SSEH algorithm are as follows. Initially, each player (data point) is an independent coalition (cluster), and CS is a coalition structure with N singleton coalitions (line 1). Then, for each player, the utility of its coalition is calculated (line 6), and whether joining other coalitions can improve its utility value is judged (line 7). If joining a new coalition can improve its utility, the player will leave the current coalition and choose the new coalition that maximizes its utility (lines 8-14). Until each player finds the coalition that maximizes its utility, the coalition structure reaches Nash stability. Finally, the obtained clusters are merged, until the ground truth number of the cluster is obtained (lines 17-20).

Algorithm 1 SSEH

```
Input: dataset $X$, set of base partition $\Pi$, similarity matrix $S$, true cluster number $K$
Output: consensus clustering $CS$
$CS = \{\{x_1\}, \{x_2\}, \dots, \{x_N\}\}$
Change = TRUE
while Change do
    Change = FALSE
    for each $x_i$ do
        Calculate the utility of its current coalition $CS_i$ using Equation (10)
        Search for a better coalition $C$ for $x_i$
        if the utility of $C$ is higher than that of $CS_i$ then
            move player $x_i$ to its new coalition $C$
            Change = TRUE
            if $CS_i$ is empty then
                delete $CS_i$
            end if
        end if
    end for
end while
while cluster number > true number $K$ do
    Find $C_i$, $C_j$ using Equation (12)
    Merge $C_i$ and $C_j$ into a new cluster for $CS$, and delete $C_i$, $C_j$ from $CS$
end while
return $CS$
```

3.3. **Subspace retrieval for the consensus partition.** In this section, we will illustrate the subspace where the consensus partition locates, i.e., retrieve the subspace for a given partition, which consists of clusters.

We set u_{pi} to 1 if x_i is assigned to the pth cluster in the partition, and then update the cluster centers using Equation (13) [10].

$$v_{pt} = \frac{(1+\eta)\sum_{i=1}^{N} u_{pi}x_{it} - \eta\sum_{i=1}^{N} x_{it}}{(1+\eta)\sum_{i=1}^{N} u_{pi} - \eta N}. \tag{13}$$

However, we cannot use Equation (14) [10] to calculate the weight vector, as does ERKM, since the value of γ is unknown for the consensus partition.

$$w_j = \frac{\exp(-D_j/\gamma)}{\sum_{l=1}^{D}\exp(-D_l/\gamma)} \tag{14}$$

where

$$D_j = (1+\eta)\sum_{p=1}^{K}\sum_{i=1}^{N} u_{pi}(x_{ij} - v_{pj})^2 - \eta\sum_{p=1}^{K}\sum_{i=1}^{N}(x_{ij} - v_{pj})^2. \tag{15}$$

We need to search the weight vector space for a feasible solution that satisfies the following constraints:

$$\sum_{j=1}^{D} w_j(x_{ij} - v_{pj})^2 - \sum_{j=1}^{D} w_j(x_{ij} - v_{qj})^2 \le 0 \text{ if } u_{pi} = 1, \forall q \ne p \tag{16}$$

$$w_j \ge 0, \quad j = 1, \dots, D \tag{17}$$

$$\sum_{j=1}^{D} w_j = 1. \tag{18}$$

The first constraint contains $(K-1)\times N$ inequalities for N objects in the dataset X with K classes. For example, Iris dataset has 150 objects with 3 classes, so the corresponding constraint (16) has 300 inequalities. Generally, there is no feasible solution that satisfies constraints (16)-(18) at the same time, so we relax the constraint (18), and describe the solution of the weight vector as the following convex optimization problem:

$$\min \left|\sum_{j=1}^{D} w_j - 1\right|_1 \tag{19}$$

$$\text{s.t.} \ \sum_{j=1}^{D} w_j(x_{ij}-v_{pj})^2 - \sum_{j=1}^{D} w_j(x_{ij}-v_{qj})^2 \le 0 \text{ if } u_{pi}=1, \ \forall q \ne p \tag{20}$$

$$w_j \ge 0, \quad j=1,\dots,D \tag{21}$$

where $|\cdot|_1$ is the L_1 norm.

Since $\mathbf{0}$ is a trivial solution that satisfies the constraints (20) and (21), the convex optimization has at least one solution. Generally, the solution $W=[w_1,w_2,\dots,w_D]$ is a point near $\mathbf{0}$. We normalize W to $\mathbf{1}$ to satisfy the constraint (18), i.e.,

$$W := \frac{W}{|W|_1}. \tag{22}$$

And then, we find that a few constraints in (16) are not satisfied. The conflicts are caused by the amplification of the left terms of these constraints, which are tiny positives approximate 0. Before normalization, the left terms are regarded as zeros. We omit these conflicts and take the weight vector W as an approximate subspace for the consensus partition, which is approximately consistent with the constraints (16)-(18). In the following experiments, we illustrate that the subspace of the consensus partition is a new subspace different from that of base partitions.

4. Experiments and Applications.

4.1. Datasets and evaluation metrics.

In order to test the proposed SSEH, we carry out experiments on 16 UCI datasets and 4 synthetic datasets [18, 27]. The detailed information of each data set is shown in Table 1. All datasets are preprocessed by column normalization. The four 2-dimensional synthetic datasets are employed for visualization of different cluster shapes, such as spherical, long curve, closely connected, or separated, as shown in Figures 1(a) and 3. The 16 UCI datasets have binary or multiple classes and various dimensions, ranging from 2 to 33, with balanced (e.g., Iris) or unbalanced class distribution (e.g., Glass).

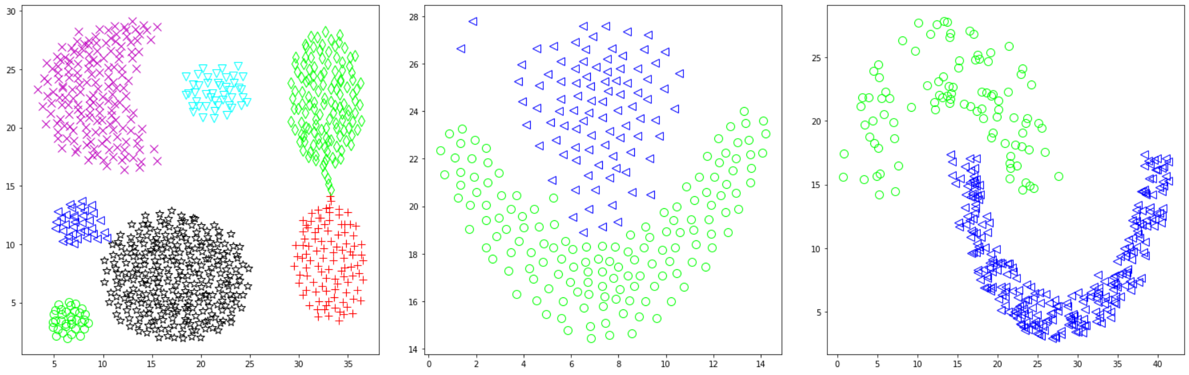
Three evaluation metrics are used to compare the experimental results: Clustering Accuracy (CA) [28], Normalized Mutual Information (NMI) [29] and Adjusted Rand Index (ARI) [30]. Clustering accuracy of clustering results is defined as

$$CA = \frac{1}{N}\sum_{i=1}^{N} \varphi(q_i, map(c_i)) \tag{23}$$

where q_i, c_i denote the the ground truth label and output label of the ith point respectively. $\varphi(x,y)=1$ if $x=y$, and 0 otherwise. The map function is used to permute clustering labels to match the ground truth labels.

TABLE 1. Information of the 20 datasets

Datasets	# Instances	# Features	# Classes	Class distribution	Source
Iris	150	4	3	50 : 50 : 50	UCI
Heart	270	13	2	120 : 150	UCI
Wine	178	13	3	48 : 59 : 71	UCI
wdbc	569	30	2	212 : 357	UCI
Seed	210	7	2	70 : 140	UCI
Diabetes	1151	19	2	540 : 611	UCI
Fertility	100	9	2	88 : 12	UCI
Pop-failure	540	20	2	46 : 494	UCI
Pima	768	8	2	268 : 500	UCI
Glass	214	9	6	29 : 76 : 70 : 17 : 13 : 9	UCI
Haberman	306	3	2	225 : 81	UCI
Spiral	312	2	3	106 : 101 : 105	[18]
Jain	373	2	2	97 : 276	[27]
Flame	240	2	2	146 : 94	[18]
Aggregation	788	2	7	34 : 36 : 170 : 45 : 271 : 102 : 130	[18]
Wpbc	194	33	2	148 : 46	UCI
Ecoli	336	7	7	143 : 77 : 2 : 2 : 35 : 20 : 57	UCI
Bupa	345	6	2	145 : 200	UCI
Breast Tissue	106	9	6	21 : 15 : 18 : 16 : 14 : 22	UCI
Transfusion	748	4	2	573 : 175	UCI



(a) Aggregation (b) Flame (c) Jain

FIGURE 3. Three 2-dimensional datasets

NMI is used to measure the average mutual information obtained by pair matching between the clustering partition results and the actual sample labels.

$$NMI = \frac{\sum_{q=1}^{Q} \sum_{c=1}^{C} N_{qc} \log \frac{N N_{qc}}{N_q N_c}}{\sqrt{\left(\sum_{q=1}^{Q} N_q \log \frac{N_q}{N} \right) \left(\sum_{c=1}^{C} N_c \log \frac{N_c}{N} \right)}} \tag{24}$$

where Q and C are the numbers of ground truth labels and the output labels, respectively. N denotes the number of data samples, N_q denotes the number of data samples in the qth ground truth label, N_c denotes the number of data samples with the cth output cluster, and N_{qc} denotes the number of data samples in which the data samples of the qth cluster in the ground truth label are classified into the cth cluster in the output label.

ARI is an extension of the Rand Index (RI), which is used to calculate the similarity between the predicted value and the true value of the sample. It is defined as follows.

$$ARI = \frac{\sum_{qc}\binom{N_{qc}}{2} - \left[\sum_q \binom{N_q}{2}\sum_c \binom{N_c}{2}\right] \Big/ \binom{N}{2}}{\frac{1}{2}\left[\sum_q \binom{N_q}{2} + \sum_c \binom{N_c}{2}\right] - \left[\sum_q \binom{N_q}{2}\sum_c \binom{N_c}{2}\right] \Big/ \binom{N}{2}} \tag{25}$$

CA, NMI and ARI all range in $[0, 1]$, the larger the values, the better the clustering results.

4.2. **Experimental setup.** We compare SSEH with three algorithms, including HGCE [14], G_EWKM, and SSEH_1. HGCE is updated by merging clusters obtained by hedonic games at the Nash equilibrium. We design G_EWKM and SSEH_1 as two versions of HGCE by replacing K-means with EWKM (Entropy Weighting K-Means) [8] and ERKM as the base partition generator respectively, keeping the rest unchanged. G_EWKM and SSEH_1 aim at evaluating the effectiveness of EWKM and ERKM as the base partition generator. SSEH uses ERKM as the base partition generator and assesses the cluster stability in similarity measurement to improve the ensemble performance. Hence, the comparison of SSEH_1 with SSEH can uncover the validity of cluster stability assessment.

Firstly, base partitions are generated by ERKM in SSEH. Concerning the feasibility of grid search in the wide range of the negative coefficient γ of ERKM, we fix the range of γ in $[0.01, 10]$, set the step to 0.05, and pick γ for ERKM. Then, ERKM generates 200 base partitions totally. In SSEH, the new pairwise similarity measurement is used, and θ is set to 0.5 as suggested [17]. ERKM has another parameter η, which affects the clustering results too; hence we test the value of $\eta = 0.02$, 0.1, and 0.2, which are the minimum, midpoint and maximum of the suggested range [10], and record the best performance for each dataset.

Similarly, HGCE, G_EWKM and SSEH_1 generate 200 base partitions by using K-means, EWKM and ERKM as the generators respectively. The three algorithms use the similarity measure based on evidence accumulation in HGCE to quantify the similarity between data points. Additionally, the cluster number K is chosen randomly from the interval $\left[2, \sqrt{N}\right]$ for the four algorithms [14]. The details are shown in Table 2.

Table 2. Comparison of the four algorithms

Algorithms	**Generators**	**Similarity assessments**
HGCE	K-means	Evidence accumulation
G_EWKM	EWKM	Evidence accumulation
SSEH_1	ERKM	Evidence accumulation
SSEH	ERKM	Cluster stability

4.3. **Experimental results.** For each pair of dataset and algorithm, the average values of CA, NMI and ARI of 30 runs are shown in Tables 3-5 respectively. The best values for each dataset are shown in bold, and the second-best ones are shown in italics. W/T/L (Wins/Ties/Losses) records the number of times that the algorithm achieves the best value/equals the best value/is not the best value.

1) W/T/L

From Tables 3-5, we observe that, in the view of W/T/L, SSEH performs significantly better than other methods on all test datasets. Specially, SSEH wins on 10, 6 and 8 datasets in terms of CA, NMI and ARI respectively, higher than that of other algorithms. Hence, SSEH is more advantageous in getting the best performances.

Table 3. CA of the four algorithms on 20 datasets

Datasets	HGCE	G_EWKM	SSEH_1	SSEH
Iris	0.8627	0.8673	*0.9582*	**0.9584**
Heart	0.7219	0.6636	**0.7384**	*0.7252*
Wine	*0.9609*	**0.9629**	0.9545	0.9446
wdbc	0.8510	0.7890	*0.8970*	**0.8981**
Seed	**0.9181**	*0.9057*	0.8663	0.8624
Diabetes	*0.5336*	**0.5369**	0.5300	0.5300
Fertility	0.5003	0.5047	*0.5057*	**0.5063**
Pop-failure	0.5768	0.5393	*0.6452*	**0.6467**
Pima	*0.6115*	0.5915	**0.6239**	0.6059
Glass	0.4402	0.4322	*0.4852*	**0.4902**
Haberman	0.5344	0.5395	*0.5815*	**0.5850**
Spiral	0.9527	0.*9797*	**1.0000**	**1.0000**
Jain	0.8131	0.7720	**0.8967**	*0.8761*
Flame	0.8146	**0.8742**	*0.8278*	0.8160
Aggregation	0.7235	0.7115	**0.7539**	*0.7310*
Wpbc	0.5838	0.5875	*0.6005*	**0.6163**
Ecoli	*0.6213*	0.5696	**0.6243**	0.5499
Bupa	0.5245	0.5346	*0.5448*	**0.5499**
Breast Tissue	0.5050	0.5280	*0.5550*	**0.5553**
Transfusion	0.5386	0.5557	*0.5578*	**0.5744**
W/T/L	1/0/19	3/0/17	5/1/14	**10/1/9**

Table 4. NMI of the four algorithms on 20 datasets

Datasets	HGCE	G_EWKM	SSEH_1	SSEH
Iris	0.7577	0.7581	*0.8594*	**0.8601**
Heart	*0.1844*	0.1295	**0.1857**	0.1680
Wine	0.8529	**0.8645**	*0.8541*	0.8361
wdbc	0.4375	0.3492	**0.5274**	*0.5267*
Seed	**0.7416**	*0.7253*	0.6485	0.6381
Diabetes	**0.0055**	*0.0051*	0.0013	0.0013
Fertility	**0.0203**	0.0186	*0.0198*	0.0197
Pop-failure	0.0029	*0.0030*	0.0028	**0.0031**
Pima	*0.0393*	0.0330	**0.0460**	0.0376
Glass	0.3581	0.3304	*0.3752*	**0.3760**
Haberman	*0.0021*	**0.0034**	0.0010	0.0009
Spiral	0.9084	*0.9654*	**1.0000**	**1.0000**
Jain	0.4477	0.3316	**0.5928**	*0.5746*
Flame	0.4295	**0.5561**	*0.4815*	0.3472
Aggregation	0.8078	0.8047	**0.8366**	*0.8141*
Wpbc	0.0237	0.0192	*0.0599*	**0.0717**
Ecoli	**0.5884**	*0.5789*	0.5754	0.5410
Bupa	0.0026	0.0049	*0.0073*	**0.0076**
Breast Tissue	0.5187	0.5170	*0.5339*	**0.5638**
Transfusion	*0.0060*	**0.0132**	0.0012	0.0010
W/T/L	4/0/16	4/0/16	5/1/14	**6/1/13**

Table 5. ARI of the four algorithms on 20 datasets

Datasets	HGCE	G_EWKM	SSEH_1	SSEH
Iris	0.6801	0.6851	*0.8810*	**0.8816**
Heart	*0.2310*	0.1596	**0.2326**	0.2143
Wine	*0.8806*	**0.8869**	0.8643	0.8372
wdbc	0.5130	0.3711	*0.6303*	**0.6331**
Seed	**0.7748**	*0.7459*	0.6548	0.6404
Diabetes	**0.0063**	*0.0060*	0.0024	0.0024
Fertility	−0.0125	−0.0125	*−0.0078*	**−0.0073**
Pop-failure	0.0005	−0.0003	*0.0085*	**0.0103**
Pima	*0.0558*	0.0413	**0.0604**	0.0445
Glass	0.1856	0.1840	*0.2371*	**0.2383**
Haberman	−0.0025	−0.0032	*0.0057*	**0.0068**
Spiral	0.8830	*0.9572*	**1.0000**	**1.0000**
Jain	0.4163	0.3205	**0.6385**	*0.5928*
Flame	0.4455	**0.5964**	*0.4517*	0.4093
Aggregation	0.6284	0.6238	**0.6890**	*0.6556*
Wpbc	0.0233	0.0211	*0.0452*	**0.0505**
Ecoli	*0.4395*	0.4095	**0.5227**	0.4298
Bupa	−0.0004	0.0028	*0.0070*	**0.0078**
Breast Tissue	0.2929	0.3266	**0.3737**	*0.3711*
Transfusion	0.0015	**0.0098**	0.0008	*0.0052*
W/T/L	2/0/18	3/0/17	6/1/13	**8/1/11**

2) Friedman test

We use Friedman rank test [31] for the statistical comparison of these techniques over the 20 datasets. Table 6 presents the average Friedman ranks summarized from Tables 3-5. Lower ranks are better, and the best performing algorithm is the one presenting the lowest average rank. SSEH_1 is the algorithm with the lowest average ranks in terms of CA, NMI and ARI (average rank = 1.9, 2.1 and 1.95 respectively), and SSEH is the second lowest one (average rank = 1.95, 2.5 and 2.1 respectively). From the view of average rank, we observe that SSEH_1 is better than the others.

Table 6. Average ranks and p-values for different metrics. Best ranks are high lighted in boldface, and the second-best ones are shown in italics.

Metric	HGCE	G_EWKM	SSEH_1	SSEH	p-value
CA	3.15	3	**1.9**	*1.95*	0.0010
NMI	*2.5*	2.9	**2.1**	*2.5*	0.3267
ARI	2.825	3.125	**1.95**	*2.1*	0.0085

To compare the four algorithms (i.e., HGCE, G_EWKM, SSEH_1 and SSEH), we evaluate the following hypothesis H_0 using Friedman test.

Null hypothesis H_0: The four algorithms do not show any significant difference when used for clustering on the datasets.

We calculate p-value for each test, and the hypothesis is checked at $\alpha = 0.05$ significance level, as shown in Table 6.

From the views of both CA and ARI, Friedman test results are significant at $\alpha = 0.05$. Thus, we reject Null hypothesis H_0. Namely, the four algorithms perform significantly different from each other at the test datasets. However, in the view of NMI, the four algorithms do not show any significant differences.

Furthermore, we conduct pairwise comparisons using Nemenyi multiple comparison test [32] for CA and ARI. We calculate p-value for each test, and the hypothesis is checked at $\alpha = 0.05$ significance level, as shown in Table 7.

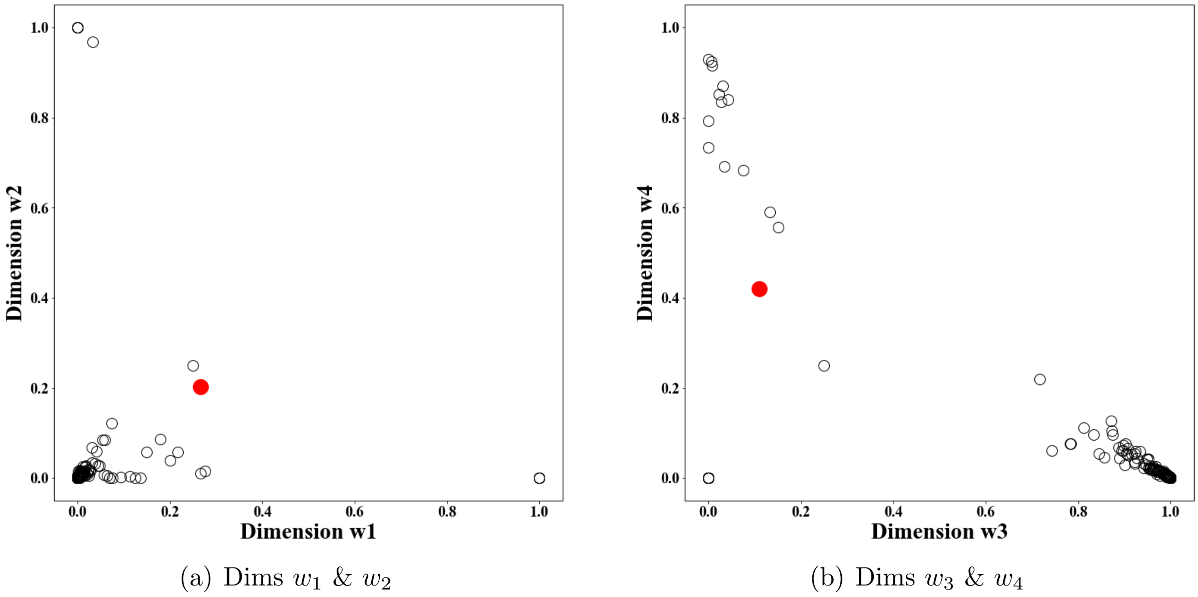
TABLE 7. p-values of using Nemenyi multiple comparison test for CA and ARI

Metric		HGCE	G_EWKM	SSEH_1
	G_EWKM	0.983	–	–
CA	SSEH_1	0.012	0.036	–
	SSEH	0.017	0.050	0.999
	G_EWKM	0.083	–	–
ARI	SSEH_1	0.140	0.021	–
	SSEH	0.285	0.058	0.983

From Table 7, we observe that SSEH and SSEH_1 have no significant difference from each other in the views of CA and ARI. However, both of them are significantly different from HGCE and G_EWKM in the view of CA, and SSEH_1 is significantly different from G_EWKM in the view of ARI.

4.4. **Subspace of the consensus partition.** We take Iris dataset as an example to search the subspace for the final partition. The ERKM is employed to generate 200 base partitions, and the parameter η is set to 0.02. Iris dataset has 4 dimensions, so we set $W = [w_1, w_2, w_3, w_4]$. Figure 4 illustrates the 200 base subspaces and the subspace corresponding to the final consensus partition. Figure 4(a) shows the dimensions w_1 and w_2, and Figure 4(b) shows the dimensions w_3 and w_4, respectively.

As shown in Figure 4, the 200 circles show the dimension weights of the base partitions. In Figure 4(a), most of the circles focus on the bottom left corner. In Figure 4(b), most of them concentrate on the bottom right corner with a few ones scattered at the



(a) Dims w_1 & w_2

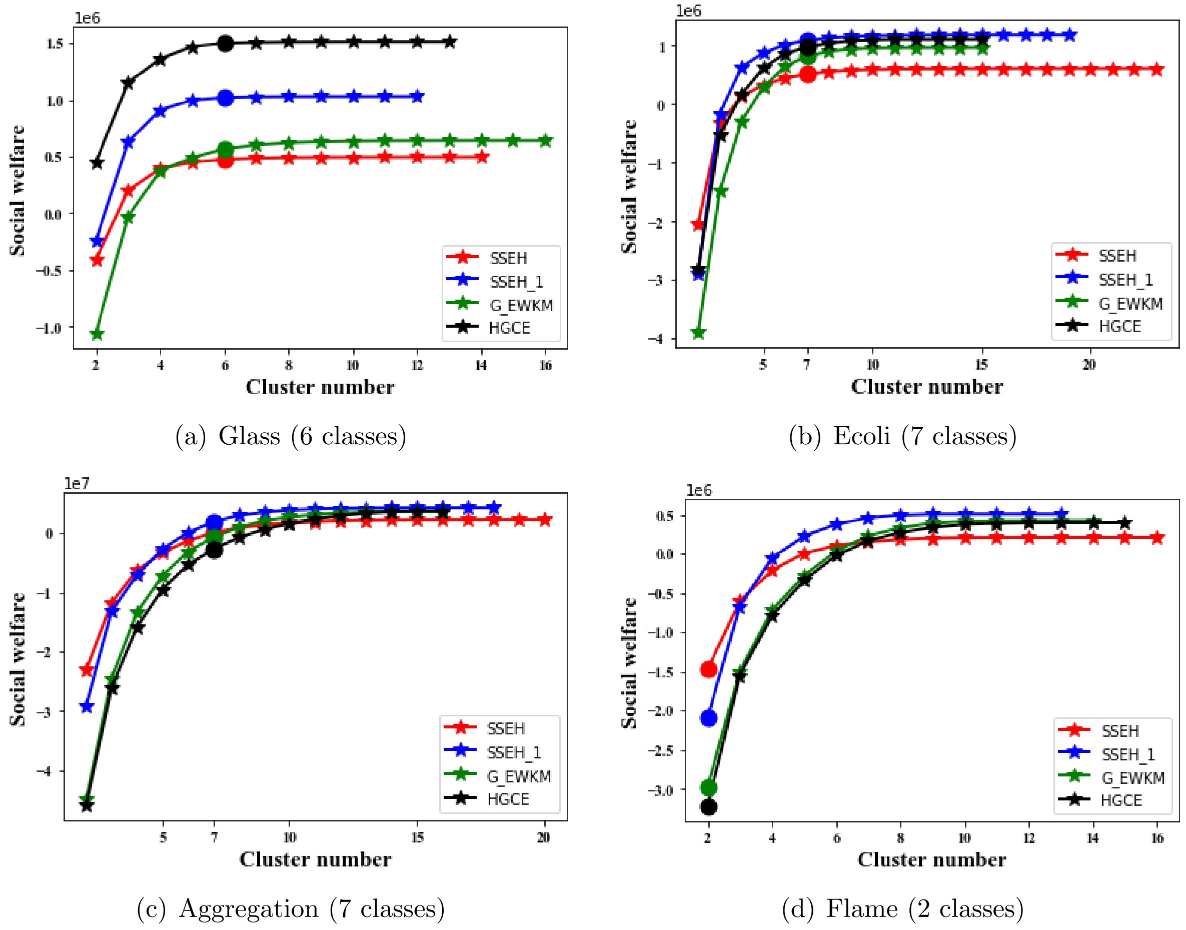
(b) Dims w_3 & w_4

FIGURE 4. Dimension weight comparison

top. The weights of the consensus partition obtained by SSEH are found by the approximate optimal solution of the problem in Section 3.3. The red dot in Figure 4 illustrates the dimension weights $[w_1, w_2, w_3, w_4]$ of the consensus partition, and the corresponding values after normalization are 0.2669, 0.2024, 0.1110, and 0.4196 respectively. Thus, we can observe that the subspace of consensus partition is different from those of the base partitions, and the SSEH explores a new subspace.

In addition, there are several special data points (6 data points in this case) which conflict with the consensus partition. Before normalization, the left terms of constraint (16) for these points are tiny positives less than 1.0e-10, which are within the error range of convex optimization, and then the convex optimization method takes W (before normalization) as a feasible solution of the subspace for the consensus partition. The existence of these conflicts shows that the subspace we find approximately describes the subspace of the consensus partition.

4.5. **Social welfare in merging process.** In the SSEH algorithm, we merge the small clusters step by step until the number of clusters equals the ground truth number. In this subsection, we record the social welfare values till merging to 2 clusters. The values of 4 datasets are illustrated in Figure 5. The merging process is shown from right to left, and the right end is the social welfare at Nash equilibrium. We observe that the algorithms reach Nash equilibrium with different cluster numbers, and the values of datasets with multiple classes show horizontal lines at the beginning of merging, and then start to decrease virtually at the ground truth number of clusters (the circle marker). For the



(a) Glass (6 classes)

(b) Ecoli (7 classes)

(c) Aggregation (7 classes)

(d) Flame (2 classes)

FIGURE 5. (color online) Merging process from right to left binary class datesets, the values decrease far from the ground truth number, as shown in Figure 5(d). Finding the ground truth number of clusters is the issue concerned by clustering validation, and the social welfare is a possible candidate for clustering validation function in the view of our observations and deserves further study.

4.6. **Applications.** Clustering ensemble has developed rapidly since it was proposed. It can be applied to text clustering to make clustering results more accurate. The text data may contain thousands of documents with multiple categories, such as the 20-Newsgroups data (http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html). The text data are typical high dimensional due to the high dimensions of the word vectors. Generally, dimension reduction techniques, such as latent semantic analysis or generalized principal component analysis, can be used to reduce the dimension. The reduction is global while each topic in the text is closely related to some special features (i.e., keywords). In real data, keywords are different from topic to topic, and the descriptive abilities of keywords are different from each other. Thus, subspace clustering is a suitable technique to find different topics in the text. We can identify several important relevant dimensions that represent important keywords, indicating the topics of the corresponding clusters [8]. Such important dimensions will be assigned high weights by the ERKM algorithm. Based on hundreds of base partitions, the consensus partition will provide more stable clusters. Retrieval of the subspace for each cluster will find important dimensions (keywords), which have larger weights than the others, and the semantic combination of these keywords will lead to more insights into the topics of the corresponding clusters.

It can also be applied to medical treatment, such as disease diagnosis, and medical image retrieval. Researchers have accumulated a wealth of molecular data, from genome to transcriptome, proteome, and epigenome by means of high-throughput technologies. Exploitation of high-throughput molecular data would significantly facilitate precision medicine and personalized treatment, which target discrete molecular subclasses of complex diseases with specific genetic or epigenetic profiles [16]. In the view of subspace clustering, the subclass of complex diseases is the cluster to be found, and the specific genetic or epigenetic profiles are the features of the disease (the subspace). Due to the high dimensionality of high-throughput molecular data, traditional clustering algorithms usually fail to find the true clusters. Ensemble clustering integrates individually generated basic partitions, and the final consensus partition can maximally agree with the basic ones, and be superior to the previous clustering algorithm. It is validated that ensemble of clustering on multiple molecular data types outperforms that of single molecular data type [16]. Similar to the text clustering ensemble, a better-quality and robust consensus partition of the high-throughput molecular data will find accurate clusters corresponding to subclasses of complex diseases with specific genetic or epigenetic profiles as the features. Weighing the features by soft subspace clustering helps to alleviate the detriment effect of irrelevant and noisy features, and understanding of various molecular data types will further improve patient stratification and precision medicine. Besides, clustering ensemble has been successfully applied in recommendation system, speech recognition and other fields [33]. Our algorithm can also be used in these fields to help find better and robust clustering results.

5. Conclusion.

In this paper, the soft subspace clustering algorithm with negative entropy, ERKM, is used to generate diversified base partitions, and the hedonic game is used to integrate the base partitions. We merge the small clusters produced by the hedonic game to recover the large true clusters and consider the cluster stability in the similarity measure of data points. The proposed SSEH_1 adopts the similarity assessment based on evidence accumulation in HGCE, and shows the lowest average ranks in terms of three metrics. Meanwhile, the proposed SSEH algorithm has advantages in finding the best partitions on test datasets. However, when the base partitions are not good enough, the clustering result is not the best one. This observation suggests that the ensemble selection is necessary for a good ensemble. As future work, we plan to study the problem of ensemble selection, and reduce the influence of poor-quality base partitions on the clustering ensemble results.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under Grant 72072154 and Grant 61773331. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the representation.

REFERENCES

[1] M. Medhat, Y. F. Hassan and A. Elsayed, Humans and bots web session identification using K-means clustering, *ICIC Express Letters*, vol.13, no.12, pp.1149-1156, 2019.

[2] Z. Deng, K. Choi, Y. Jiang et al., A survey on soft subspace clustering, *Information Sciences*, vol.348, pp.84-106, 2016.

[3] H. Kriegel, P. Krger and A. Zimek, Subspace clustering, *Wiley Interdiplinary Reviews: Data Mining and Knowledge Discovery*, vol.2, no.4, pp.351-364, 2012.

[4] K. Sim, V. Gopalkrishnan, A. Zimek and G. Cong, A survey on enhanced subspace clustering, *Data Mining and Knowledge Discovery*, vol.26, no.2, pp.332-397, 2013.

[5] E. Hancer, B. Xue and M. Zhang, A survey on feature selection approaches for clustering, *Artificial Intelligence Review*, vol.53, no.2, pp.4519-4545, 2020.

[6] R. C. D. Amorim, A survey on feature weighting based k-means algorithms, *Journal of Classification*, vol.33, no.2, pp.210-242, 2016.

[7] J. H. Friedman and J. J. Meulman, Clustering objects on subsets of attributes, *Journal of the Royal Statistical Society Series B – Statistical Methodology*, vol.66, no.4, pp.815-849, 2004.

[8] L. Jing, M. K. Ng and J. Z. Huang, An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data, *IEEE Trans. Knowledge and Data Engineering*, vol.19, no.8, pp.1026-1041, 2007.

[9] Z. Deng, K. S. Choi, F. L. Chung et al., Enhanced soft subspace clustering integrating within-cluster and between-cluster information, *Pattern Recognition*, vol.43, no.3, pp.767-781, 2010.

[10] L. Xiong, C. Wang, X. Huang et al., An entropy regularization k-means algorithm with a new measure of between-cluster distance in subspace clustering, *Entropy*, vol.21, no.7, 2019.

[11] E. Chitsaz and M. Z. Jahromi, A novel soft subspace clustering algorithm with noise detection for high dimensional datasets, *Soft Computing*, vol.20, no.11, pp.4463-4472, 2016.

[12] A. Bagherinia, B. Minaei-Bidgoli, M. Hossinzadeh et al., Reliability-based fuzzy clustering ensemble, *Fuzzy Sets and Systems*, vol.413, pp.1-28, 2021.

[13] H. S. Yoon, S. Y. Ahn, S. H. Lee et al., Heterogeneous clustering ensemble method for combining different cluster results, *Workshop on Data Mining for Biomedical Applications*, 2006.

[14] N. C. Sandes and L. V. C. André, Clustering ensembles: A hedonic game theoretical approach, *Pattern Recognition*, vol.81, pp.95-111, 2018.

[15] J. Wu, H. Liu, H. Xiong et al., K-means-based consensus clustering: A unified view, *IEEE Trans. Knowledge and Data Engineering*, vol.27, no.1, pp.155-169, 2015.

[16] H. Liu, R. Zhao, H. Fang et al., Entropy-based consensus clustering for patient stratification, *Bioinformatics*, vol.33, no.17, pp.2691-2698, 2017.

[17] D. Huang, C. Wang and J. Lai, Locally weighted ensemble clustering, *IEEE Trans. Cybernetics*, vol.48, no.5, pp.1460-1473, 2018.

[18] C. Zhong, L. Hu, X. Yue et al., Ensemble clustering based on evidence extracted from the co-association matrix, *Pattern Recognition*, vol.92, pp.93-106, 2019.

[19] M. Mojarad, H. Parvin, S. Nejatian et al., Consensus function based on clusters clustering and iterative fusion of base clusters, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol.27, no.1, pp.97-120, 2019.

[20] A. Strehl and J. Ghosh, Cluster ensembles – A knowledge reuse framework for combining multiple partitions, *Journal of Machine Learning Research*, vol.3, pp.583-617, 2002.

[21] G. Chalkiadakis, E. Elkind and M. Wooldridge, *Computational Aspects of Cooperative Game Theory*, Morgan & Claypool Publishers, 2011.

[22] V. K. Garg, Y. Narahari and M. N. Murty, Novel Biobjective Clustering (BiGC) based on cooperative game theory, *IEEE Trans. Knowledge and Data Engineering*, vol.25, no.5, pp.1070-1082, 2013.

[23] M. Feldman, L. Lewin-Eytan and J. Naor, Hedonic clustering games, *ACM Trans. Parallel Computing*, vol.2, no.1, pp.1-48, 2015.

[24] A. Nazari, A. Dehghan, S. Nejatian et al., A comprehensive study of clustering ensemble weighting based on cluster quality and diversity, *Pattern Analysis and Applications*, vol.22, pp.133-145, 2019.

[25] S. Abbasi, S. Nejatian, H. Parvin et al., Clustering ensemble selection considering quality and diversity, *Artificial Intelligence Review*, vol.52, pp.1311-1340, 2019.

[26] E. N. Barron, *Game Theory: An Introduction*, 2nd Edition, John Wiley & Sons, 2013.

[27] A. K. Jain and M. Law, Data clustering: A user's dilemma, *Proc. of the 1st International Conference on Pattern Recognition and Machine Intelligence (PReMI'05)*, DOI: 10.1007/11590316_1, 2005.

[28] C. Lu, J. Feng, Z. Lin et al., Subspace clustering by block diagonal representation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.41, no.2, pp.487-501, 2019.

[29] M. Liu, Y. Wang, J. Sun et al., Structured block diagonal representation for subspace clustering, *Applied Intelligence*, pp.1-14, 2020.

[30] L. Hubert and P. Arabie, Comparing partitions, *Journal of Classification*, vol.2, no.1, pp.193-218, 1985.

[31] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, *Journal of the American Statistical Association*, vol.32, no.200, pp.675-701, 1937.

[32] P. B. Nemenyi, *Distribution-Free Multiple Comparisons*, Ph.D. Thesis, Princeton University, 1963.

[33] R. Amami, G. Manita and A. Smiti, Robust speech recognition using consensus function based on multi-layer networks, *The 9th Iberian Conference on Information Systems and Technologies (CISTI)*, pp.1-6, 2014.