

# Entropy Based Cluster Selection



Arko Banerjee, Arun K. Pujari, Chhabi Rani Panigrahi, and Bibudhendu Pati

**Abstract** Clustering has emerged as a method of unsupervised partitioning of a given set of data instances into a number of groups (called clusters) so that instances in the same group are more similar among each other with respect to instances in other groups. But there does not exist a universal clustering algorithm that can yield satisfactory result for any dataset. In this work we consider an ensemble (collection) of clusterings (partitions) of a dataset obtained in different ways and devise two methods that judiciously select clusters from different clusterings in the ensemble to construct a robust clustering. The superior performances of the proposed methods over well-known existing clustering algorithms on several benchmark datasets are empirically reported.

**Keywords** Clustering · Clustering ensemble · Consensus clustering · Entropy

## 1 Introduction

Cluster analysis or clustering [1] is a machine learning technique that unsupervisedly recognizes groups (clusters) of *similar* data instances collected from fields such as marketing, social network, bio-medical etc. [2]. In the literature numerous clustering

---

A. Banerjee (✉)

College of Engineering and Management, Kolaghat, West Bengal, India

e-mail: [arko.banerjee@gmail.com](mailto:arko.banerjee@gmail.com)

Biju Patnaik University of Technology, Rourkela, Odisha, India

A. K. Pujari

Central University of Rajasthan, Ajmer, Rajasthan, India

e-mail: [arun.k.pujari@gmail.com](mailto:arun.k.pujari@gmail.com)

C. R. Panigrahi · B. Pati

Rama Devi Women's University, Bhubaneswar, Odisha, India

e-mail: [panigrahichhabi@gmail.com](mailto:panigrahichhabi@gmail.com)

B. Pati

e-mail: [patibibudhendu@gmail.com](mailto:patibibudhendu@gmail.com)

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2021  
C. R. Panigrahi et al. (eds.), *Progress in Advanced Computing and Intelligent Engineering*,  
Advances in Intelligent Systems and Computing 1299,  
[https://doi.org/10.1007/978-981-33-4299-6\\_26](https://doi.org/10.1007/978-981-33-4299-6_26)

algorithms [3] have been proposed that conceptually differ among themselves for clustering datasets having a different distribution of instances. But since there does not exist a universal clustering algorithm that can provide an acceptable result for any dataset [4], clustering ensemble (also referred to as consensus clustering) [5] techniques have been proposed in the literature that optimally combine the results of different clusterings to yield a qualitatively better and robust clustering solution. But since finding a consensus clustering in the space of clusterings is an NP-hard problem, several heuristics are being proposed to find an acceptable consensus, which can be broadly categorized into hyper-graph based, information theory based, mixture model (EM) based, voting based and co-association based methods.

In this work, we propose a new way of arriving at a consensus clustering from an ensemble of clusterings with a true number of clusters. We present a concept of consensus clustering that can be formed by selecting clusters from the existing clusterings in the ensemble. Every cluster in the ensemble can be assigned with an entropy measure representing its reliability in the ensemble. A cluster having lower entropy can be considered being of higher priority in forming a consensus. Hence selection of clusters and overlapping problem (if any) of selected clusters can be resolved by prioritizing clusters having lower entropy values to form the final consensus clustering. In a second approach we propose to select a cluster having the lowest entropy from an ensemble of clusterings and iteratively the process is repeated after removing the data instances of the selected clusters to construct the final clustering. In the empirical section, we explain the significance of our proposed methods and show their superiority over well-known existing clustering methods on several benchmark datasets.

The outline of the rest of this paper is as follows. The related work is written in Sect. 2. Section 3 presents the problem statement and the proposed methods are described in Sect. 4. Section 5 empirically demonstrates the performance of the proposed algorithms and Sect. 6 concludes the paper.

## 2 Related Work

In recent years a significant number of research papers have been published on new algorithms of consensus clustering [7]. We give a brief overview of the well-known clustering ensemble algorithms in this section.

Fred et al. [5] combined ensemble of clusterings produced in different ways into a co-association matrix (also called similarity matrix) and the final consensus clustering was obtained by applying hierarchical single-link algorithm on the co-association matrix. The method was referred to as Evidence Accumulation (EAC) method. Huang et al. [8] applied average-linked agglomerative clustering technique on a cluster-level weighted co-association matrix to derive a consensus clustering. The proposed method was referred by them as Locally Weighted Ensemble Accumulation (LWEA) method, in which the said cluster-level weight depended on a user defined parameter, and the best result was reported in the experiment. Strehl et al. [6] represented the

problem of clustering ensemble as a combinatorial optimization problem in terms of shared mutual information and tried to solve the problem by representing the clustering ensemble as a hyper-graph, where each clustering is an hyper-edge. They proposed three different algorithms, namely Cluster-based Similarity Partitioning (CSPA), Hyper Graph Partitioning (HGPA), and Meta-Clustering (MCLA). Huang et al. [10] implemented Crowd Agreement Estimation and Multigranularity Link Analysis to solve the clustering ensemble problem. They proposed two algorithms namely, weighted evidence accumulation clustering (WEAC) and graph partitioning with multi-granularity link analysis (GP-MGLA). Huang et al. [10] computed Probability Trajectory based Similarity between clusters in the clusterings and proposed two algorithms namely, Probability Trajectory Accumulation(PTA) and Probability Trajectory Based Graph Partitioning(PTGP). PTA was an agglomerative clustering algorithm that depended on two user-defined parameters. In our experimental comparison the best result of the method is reported. Nguyen et al. [12] presented iterative voting techniques to find a clustering ensemble in which the principal operation was to learn the closest cluster from each data instance in every iteration. On this basis they proposed their algorithms, namely Iterative Voting Consensus (IVC), Iterative Pairwise Consensus (IPC) and Iterative Probabilistic Voting Consensus (IPVC). Fern et al. [9] proposed Cluster and select(CAS) method that used NMI [6] measure to group similar clusterings from the initial ensemble. The final ensemble was created by selecting a representative from each group of clusterings. Finally, CSPA was used as the clustering ensemble method on the final ensemble. The size of the final ensemble was a user-defined parameter, and the best result was reported in the experiment.

### 3 Problem Statement

Let  $D$  be the set of  $N$  data instances  $D = \{d_1, d_2, \dots, d_N\}$ . We are given with an ensemble of  $M$  clusterings,  $E = \{P_1, P_2, \dots, P_M\}$  of  $D$ , where a clustering  $P_t$  ( $t = 1, 2, \dots, M$ ) on  $D$  is defined as a set of clusters  $P_t = \{C_1^t, C_2^t, \dots, C_K^t\}$  such that  $C_i^t \subseteq D$ ,  $C_i^t \cap C_j^t = \phi, (i \neq j)$  and  $\bigcup_{i=1}^K C_i^t = D$ , ( $i, j = 1, 2, \dots, K$ ). The goal of clustering ensemble is to find a clustering  $P^* = \{C_1^*, C_2^*, \dots, C_k^*\}$  that is a sort of median of the given ensemble of clusterings  $P_1, P_2, \dots$  and  $P_M$ .  $P^*$  is called the consensus clustering.

In this work we approach the problem of consensus clustering by assigning an entropy value to individual cluster in the ensemble. To do that let us define the joint probability  $p(C_i^t, C_j^s)$  that signifies the level of agreement cluster  $C_i^t$  ( in clustering  $P_t$  ) has with cluster  $C_j^s$  (in clustering  $P_s$ ). By level of agreement we mean the number of elements both clusters have in common. Then,

$$p(C_i^t, C_j^s) = \frac{|C_i^t \cap C_j^s|}{|C_i^t|} \quad (1)$$

Let the entropy measure  $\text{Ent}(C_i^t, C_j^s)$  represents the entropy of a cluster  $C_i^t$  w.r.t.  $C_j^s$  (and vice versa) and is computed as in Huang et al. [8] (it is to note that, our proposed methods do not depend on a specific surprisal measure),

$$\text{Ent}(C_i^t, C_j^s) = -p(C_i^t, C_j^s) \cdot \log_2 p(C_i^t, C_j^s) \quad (2)$$

Then the total entropy of the cluster  $C_i^t$  w.r.t all clusters in the ensemble  $E$  is

$$\text{Ent}(C_i^t) = - \sum_{\substack{s=1 \\ s \neq t}}^M \sum_{j=1}^K p(C_i^t, C_j^s) \cdot \log_2 p(C_i^t, C_j^s) \quad (3)$$

Less the total entropy of a cluster more it agrees with other intersecting clusters in the ensemble. Hence a clustering, containing clusters having minimum possible entropy values, is a sort of median of the entire ensemble which approximates the consensus clustering  $P^*$ . In our work we propose two heuristics to achieve the said goal. It is also to note that, Eq. 3 tends to allot lower entropy to finer clusters than that of coarser clusters. In a trivial case, a cluster having a single data instance has an entropy value of zero.

## 4 Proposed Methods

In our first approach we put all the clusters of each clustering in the ensemble into a set say  $C$ . Since each clustering contains  $K$  clusters and there are total  $M$  number of clusterings in the ensemble  $E$ , there will be  $K \cdot M$  number of clusters in  $C$ . We consider a collection, say  $S$ , of all possible subsets of  $C$ , where each set consists of  $K$  clusters and covers all the data instances in  $D$ . Cluster overlapping may happen in a set if all the  $K$  clusters are not mutually exclusive. Out of all the sets in  $S$  we consider the set, say  $\overline{C}^*$ , in which the sum of entropy values of all the constituent clusters is minimum. If all the  $K$  clusters in  $\overline{C}^*$  are mutually exclusive, then  $\overline{C}^*$  is the desired consensus clustering  $P^*$ . Else the cluster overlapping problem is solved by prioritizing clusters having lower entropy value. More precisely, if two clusters are not disjoint, the cluster having less Entropy value will be retained, whereas the intersection part of the said two clusters is removed from the other cluster. It is to note that, reduction of a cluster reduces its degree of disagreement with other clusters in the ensemble; as a result the reduced cluster tends to have a lower entropy value than that of its previous version. The final set, without overlapping clusters, is the desired consensus clustering  $P^*$ . The proposed solution of the overlapping problem can actually decrease the entropy of the clustering (which is the sum of the total entropy values of its constituent clusters), hence resulting in a better consensus. We write down the said heuristics in form of an algorithm and we call it Entropy based Cluster Selection (ECS) Consensus Clustering. The output of the algorithm cannot be empty, as the existing clustering with the lowest entropy in the ensemble

is also a possible output. Such output can occur only when the selected clusters are non-overlapping. The time complexity of the proposed algorithm can be reduced by solving it using dynamic programming.

---

**Algorithm 1** Entropy Based Cluster Selection (ECS)
 

---

Input Ensemble of Clusterings  $E = \{P_1, P_2, \dots, P_M\}$   
 Output Consensus Clustering  $P^*$  with  $K$  clusters

- 1:  $C = \{C_i^t : C_i^t \in P_t, t = 1 \dots M, i = 1, \dots, K\}$
- 2:  $S = \{\bar{C} \subseteq C : |\bar{C}| = K \wedge \bigcup_{j=1}^K C_{i_j}^{t_j} \subseteq D\}$   
 where,  $\bar{C} = \{C_{i_j}^{t_j} : C_{i_j}^{t_j} \in C, j = 1, \dots, K, i_j \in \{1, \dots, K\}, t_j \in \{1, \dots, M\}\}$
- 3:  $\bar{C}^* \leftarrow \arg \min_{\bar{C}} \left( Ent(\bar{C}) = \sum_{j=1}^K Ent(C_{i_j}^{t_j}) : \bar{C} \in S \right)$   
 $\{Ent(.) \text{ defined in Eq. 3}\}$
- 4: *WOLOG* Let  $\bar{C}^* = \{C_1^*, C_2^*, \dots, C_K^*\} \ni Ent(C_u^*) \leq Ent(C_{u+1}^*),$   
 $u = 1, 2, \dots, K - 1$
- 5:  $T \leftarrow C_1^*$
- 6: **for**  $2 \leq l \leq K$  **do**
- 7:  $C_l^* \leftarrow C_l^* \setminus T$
- 8:  $T \leftarrow T \cup C_l^*$
- 9: **end for**
- 10:  $P^* \leftarrow \bar{C}^*$

---

We propose an alternative approach in generating a robust and good quality clustering from an ensemble of clusterings. In our second approach we follow the following steps to come up with the final clustering.

1. Consider the set of all clusters  $C$  in an ensemble of clusterings each having  $K$  number of clusters.
2. Select the cluster in  $C$  having the lowest entropy value.
3. Remove the data instances contained in the selected cluster from the dataset  $D$ .
4. Generate a new ensemble of clusterings each having  $K = K - 1$  number of clusters.
5. Repeat steps 1–4 until the remaining data instances in  $D$  is grouped into a single cluster.

The above method cannot be categorized as a traditional consensus clustering method. It iteratively generates the final clustering from a sequence of an ensemble of clusterings with a reducing number of clusters. In this method the quality of the final clustering depends more on the ensemble generation process than ECS. We write down this method in form of an algorithm and we call it Maximum Entropy Cluster Selection (MECS) Clustering.

## 5 Experimental Result

In this section, we empirically show the efficiency of our proposed methods over other existing well-known clustering ensemble algorithms. First, we discuss the data sets and the basic settings of our experiments used in the evaluation process.

### 5.1 Datasets

In our experiments, we use nine well known real-world labeled datasets, namely, *Iris*, *Wine*, *Glass*, *Image Segmentation (IS)*, *Ecoli* and *Steel Plates Faults (SPF)*. Table 1 displays the details of the datasets. All the datasets are taken from the UCI machine learning repository [11].

---

#### Algorithm 2 Maximum Entropy Cluster Selection (MECS)

---

Input Dataset  $D$  and number of desired clusters  $K$

Output Consensus Clustering  $P^*$  with  $K$  clusters

- 1: Initialization:  $P^* \leftarrow \emptyset$
  - 2: **for** each  $K \geq l \geq 2$  **do**
  - 3:   Generate Ensemble of Clusterings  $E^l = \{P_1^l, P_2^l, \dots, P_M^l\}$  on  $D$   
    where,  $P_i^l = \{C_1^i, C_2^i, \dots, C_l^i\} \ni C_i^i \subseteq D, C_i^i \cap C_j^i = \phi, (i \neq j)$   
     $\wedge \bigcup_{i=1}^l C_i^i = D, (i, j = 1, 2, \dots, l)$
  - 4:    $C_l \leftarrow \arg \min_{C_i^i} (Ent(C_i^i) : C_i^i \in P_i^l, i = 1, \dots, l, t = 1, \dots, M)$   
     $\{Ent(.) \text{ defined in Eq. 3}\}$
  - 5:    $D \leftarrow D \setminus C_l$
  - 6:    $P^* \leftarrow \{P^*, C_l\}$
  - 7: **end for**
  - 8:  $C_1 \leftarrow D$
  - 9:  $P^* \leftarrow \{P^*, C_1\}$
- 

**Table 1** Description of datasets

| Dataset      | #Instances | #Attribute | #Class |
|--------------|------------|------------|--------|
| <i>Iris</i>  | 150        | 4          | 3      |
| <i>Wine</i>  | 178        | 13         | 3      |
| <i>Glass</i> | 214        | 10         | 7      |
| <i>IS</i>    | 2310       | 19         | 7      |
| <i>Ecoli</i> | 336        | 8          | 8      |
| <i>SPF</i>   | 1941       | 27         | 7      |

**Table 2** Comparison of performance with NMI

| Method  | Iris    |          |         | Wine    |          |          | Glass    |          |          |
|---------|---------|----------|---------|---------|----------|----------|----------|----------|----------|
|         | $E/20$  | $E_d/10$ | $E_d/5$ | $E/20$  | $E_d/10$ | $E_d/5$  | $E/20$   | $E_d/10$ | $E_d/5$  |
| ECS     | 0.7515  | 0.59412  | 0.59201 | 0.4287  | 0.40556  | 0.40125  | 0.36227  | 0.34828  | 0.34356  |
| MECS    | 0.7462  | 0.57227  | 0.56718 | 0.3988  | 0.38917  | 0.3912   | 0.35986  | 0.34231  | 0.34029  |
| LWEA    | 0.74944 | 0.53154  | 0.54595 | 0.41973 | 0.39167  | 0.39726  | 0.36122  | 0.34839  | 0.34341  |
| EAC     | 0.74944 | 0.53154  | 0.542   | 0.41973 | 0.39167  | 0.39714  | 0.36182  | 0.34821  | 0.34261  |
| WEAC-AL | 0.74998 | 0.53514  | 0.55392 | 0.4287  | 0.39167  | 0.39445  | 0.36182  | 0.33849  | 0.34143  |
| GP-MGLA | 0.74998 | 0.53514  | 0.55392 | 0.4287  | 0.39167  | 0.39273  | 0.36114  | 0.35165  | 0.33545  |
| PTA-AL  | 0.49068 | 0.22804  | 0.34885 | 0.26771 | 0.22225  | 0.23623  | 0.35885  | 0.34679  | 0.33576  |
| PTA-CL  | 0.49068 | 0.2024   | 0.32519 | 0.21513 | 0.15742  | 0.23529  | 0.35684  | 0.34824  | 0.34253  |
| PTA-SL  | 0.413   | 0.30307  | 0.3361  | 0.29405 | 0.21686  | 0.24043  | 0.34524  | 0.31615  | 0.32587  |
| PTGP    | 0.68158 | 0.53225  | 0.54479 | 0.40422 | 0.39555  | 0.39867  | 0.36501  | 0.35776  | 0.34848  |
| CSPA    | 0.64280 | 0.50197  | 0.51379 | 0.38122 | 0.34395  | 0.30911  | 0.34046  | 0.32114  | 0.32071  |
| HGPA    | 0.59815 | 0.46718  | 0.47819 | 0.35480 | 0.32012  | 0.28769  | 0.31687  | 0.29889  | 0.29848  |
| MCLA    | 0.67935 | 0.53052  | 0.54302 | 0.40290 | 0.36351  | 0.32670  | 0.35983  | 0.33941  | 0.33895  |
| IVC     | 0.62025 | 0.48437  | 0.49578 | 0.36785 | 0.33189  | 0.29828  | 0.32853  | 0.30988  | 0.30946  |
| IPVC    | 0.66308 | 0.53205  | 0.52568 | 0.39324 | 0.36454  | 0.31626  | 0.35120  | 0.34037  | 0.32812  |
| IPC     | 0.6678  | 0.52151  | 0.53378 | 0.39606 | 0.35734  | 0.32115  | 0.35372  | 0.33364  | 0.33319  |
| CAS     | 0.64778 | 0.50407  | 0.51502 | 0.38418 | 0.34540  | 0.30985  | 0.34311  | 0.32249  | 0.32146  |
| Method  | IS      |          |         | Ecoli   |          |          | SPF      |          |          |
|         | $E/20$  | $E_d/10$ | $E_d/5$ | $E/20$  | $E_d/10$ | $E_d/5$  | $E/20$   | $E_d/10$ | $E_d/5$  |
| ECS     | 0.54861 | 0.51435  | 0.45085 | 0.55961 | 0.55415  | 0.55427  | 0.10418  | 0.09202  | 0.08059  |
| MECS    | 0.54275 | 0.51249  | 0.44988 | 0.55323 | 0.55172  | 0.55283  | 0.10386  | 0.09179  | 0.08021  |
| LWEA    | 0.55191 | 0.51723  | 0.45754 | 0.55608 | 0.55225  | 0.55393  | 0.10377  | 0.08482  | 0.080289 |
| EAC     | 0.53113 | 0.51381  | 0.43187 | 0.55855 | 0.55287  | 0.55426  | 0.10421  | 0.092756 | 0.08176  |
| WEAC-AL | 0.49775 | 0.47101  | 0.42107 | 0.54755 | 0.54024  | 0.531045 | 0.10336  | 0.093972 | 0.08349  |
| GP-MGLA | 0.54541 | 0.50775  | 0.44019 | 0.54085 | 0.54016  | 0.53679  | 0.10513  | 0.091142 | 0.072813 |
| PTA-AL  | 0.54527 | 0.50486  | 0.4484  | 0.51711 | 0.51647  | 0.51454  | 0.10434  | 0.093604 | 0.073398 |
| PTA-CL  | 0.55638 | 0.50981  | 0.44589 | 0.51195 | 0.50937  | 0.50806  | 0.1058   | 0.09519  | 0.073283 |
| PTA-SL  | 0.36147 | 0.36147  | 0.34005 | 0.54009 | 0.54663  | 0.54719  | 0.088353 | 0.076516 | 0.064218 |
| PTGP    | 0.54559 | 0.50305  | 0.44856 | 0.52732 | 0.52449  | 0.52489  | 0.10635  | 0.090923 | 0.079137 |
| CSPA    | 0.51454 | 0.47442  | 0.42303 | 0.49731 | 0.53647  | 0.60210  | 0.10030  | 0.08575  | 0.07463  |
| HGPA    | 0.47888 | 0.44154  | 0.39372 | 0.46285 | 0.49929  | 0.56037  | 0.09335  | 0.07981  | 0.06946  |
| MCLA    | 0.54380 | 0.50140  | 0.44709 | 0.52559 | 0.56698  | 0.63634  | 0.10600  | 0.09063  | 0.07888  |
| IVC     | 0.49650 | 0.45779  | 0.40820 | 0.47987 | 0.51766  | 0.58099  | 0.09678  | 0.08274  | 0.07202  |
| IPVC    | 0.53077 | 0.50282  | 0.43281 | 0.51299 | 0.55339  | 0.62108  | 0.10630  | 0.09088  | 0.07910  |
| IPC     | 0.53457 | 0.49289  | 0.43950 | 0.51667 | 0.55735  | 0.62554  | 0.10420  | 0.08909  | 0.07754  |
| CAS     | 0.51854 | 0.47642  | 0.42403 | 0.50118 | 0.54064  | 0.60678  | 0.10072  | 0.08611  | 0.07495  |

## 5.2 Ensemble Generation

The clusterings in the ensemble are generated by applying the K-means algorithm [1] with different random initializations. The K-means method is considered here as it is widely used in the clustering ensemble studies in the literature. We generate an ensemble of 100 clusterings for all datasets in which clusterings may repeat. We denote  $E$  as the entire ensemble,  $E/x$  as the ensemble formed by randomly selected  $x$  percentage of clusterings from  $E$ ,  $E_d$  as the distinct partitions in  $E$ , and  $E_d/x$  would mean a randomly selected  $x$  percentage clusterings from  $E_d$ . We perform tests on three types of ensembles  $E/20$ ,  $E_d/10$  and  $E_d/5$ . We run all the clustering ensemble algorithms, discussed in Sect. 2, ten times for each dataset and report the average performance. The performances (validity) of the methods are measured with the help of Normalized Mutual Information (NMI) [6], by comparing their result with the ground truth information available with each dataset. The range of NMI is between 0 and 1 and a larger value indicates better quality clustering w.r.t ground-truth information. Let  $P^*$  be the consensus clustering and  $G$  is the ground-truth clustering. The NMI score of  $P^*$  given  $G$  is defined as follows:

$$\text{NMI}(P^*, G) = \frac{\sum_{i=1}^K \sum_{j=1}^K N_{ij} \log \frac{N_{ij}N}{N_i^{P^*} N_j^G}}{\sqrt{\sum_{i=1}^K N_i^{P^*} \log \frac{N_i^{P^*}}{N} \sum_{j=1}^K N_j^G \log \frac{N_j^G}{N}}}, \quad (4)$$

where  $P^*$  and  $G$  have  $K$  (true) number of clusters and  $N$  is the total number of instances in the data set.  $N_i^{P^*}$ ,  $N_j^G$  and  $N_{ij}$  are the numbers of data instances in the  $i$ th cluster of  $P^*$ ,  $j$ th cluster of  $G$  and in both  $i$ th cluster in  $P^*$  and  $j$ th cluster in  $G$ , respectively.

## 5.3 Evaluation

We report the experimental outcomes in Table 2. The proposed *ECS* achieves highest NMI scores on the *Iris*, *Wine* and *Ecoli* datasets. It performs second best and third best in the case of *Glass* and *IS* datasets, respectively. The performance of the proposed *MECS* is inferior to that of *ECS*, but superior to most of the well-known methods.

## 6 Conclusion

In this work, we introduce the concept of cluster selection based consensus clustering technique. We demonstrate that judiciously selecting a set of clusters from the ensemble can result in a better consensus. Evaluations on different datasets show the proposed approaches are efficient and effective in improving the quality of consensus



as compared to the existing approaches when the true number of clusters is considered. In the future, we would like to explore other possibilities towards consensus using our proposed concept of cluster selection.

## References

1. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Engle wood Cliffs (1988)
2. Aggarwal, C. C. , Reddy, C.K.: Data Clustering: Algorithms and Applications, 1st ed. Chapman and Hall, London (2013). ISBN 9781466558212
3. Xu, D., Tian, Y.A.: Comprehensive survey of clustering algorithms. *Ann. Data. Sci.* **2**, 165–193 (2015)
4. Kleinberg, J.: An impossibility theorem for clustering. In: Proceedings of Advanced in Neural Information Processing Systems (2002)
5. Fred, A.L.N. , Roli, F., Kittler, J.: Finding consistent clusters in data partitions. In: Proceedings of 3rd International Workshop on Multiple Classifier Systems, vol. 2364, pp. 309–318 (2002)
6. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **3**, 583–617 (2002). <https://doi.org/10.1162/153244303321897735>
7. Zhou, Z.: Ensemble Methods: Foundations and Algorithms. CRC Press, Boca Raton, FL, USA (2012)
8. Huang, D., Wang, C., Lai, J.: Locally weighted ensemble clustering. *IEEE Trans. Cybern.* **48**(5), 1460–1473 (2018)
9. Fern, X.Z., Lin, W.: Cluster ensemble selection. *Statistical Anal. Data Mining* **1**(3), 128–141 (2008)
10. Huang, D., Lai, J., Wang, C.: Combining multiple clusterings via crowd agreement estimation and multi-granularity link analysis. *Neurocomputing* **170**, 240–250 (2015)
11. Dua, D., Graff, C.: UCI Machine learning repository. University of California, School of Information and Computer Science, Irvine, CA (2019). <https://archive.ics.uci.edu/ml>
12. Nguyen, N., Caruana, R.: Consensus clusterings. In: Proceedings of IEEE International Conference on Data Mining (ICDM), pp. 607–612 (2007). <https://doi.org/10.1109/ICDM.2007.73>