




# AutoCluster: Meta-learning Based Ensemble Method for Automated Unsupervised Clustering

Yue Liu<sup>1,2,3</sup>(✉) , Shuang Li<sup>1</sup>, and Wenjie Tian<sup>1</sup>

<sup>1</sup> School of Computer Engineering and Science, Shanghai University, Shanghai, China  
{yueliu, aimeeli, twenjie}@shu.edu.cn

<sup>2</sup> Shanghai Institute for Advanced Communication and Data Science,  
Shanghai, China

<sup>3</sup> Shanghai Engineering Research Center of Intelligent Computing System,  
Shanghai 200444, China

**Abstract.** Automated clustering automatically builds appropriate clustering models. The existing automated clustering methods are widely based on meta-learning. However, it still faces specific challenges: lacking comprehensive meta-features for meta-learning and general clustering validation index (CVI) as objective function. Therefore, we propose a novel automated clustering method named AutoCluster to address these problems, which is mainly composed of Clustering-oriented Meta-feature Extraction (CME) and Multi-CVIs Clustering Ensemble Construction (MC<sup>2</sup>EC). CME captures the meta-features from spatial randomness and different learning properties of clustering algorithms to enhance meta-learning. MC<sup>2</sup>EC develops a collaborative mechanism based on clustering ensemble to balance the measuring criterion of different CVIs and construct more appropriate clustering model for given datasets. Extensive experiments are conducted on 150 datasets from OpenML to create meta-data and 33 test datasets from three clustering benchmarks to validate the superiority of AutoCluster. The results show the superiority of AutoCluster for building an appropriate clustering model compared with classical clustering algorithms and CASH method.

**Keywords:** Clustering · Automated machine learning · Meta-learning · Model selection · Clustering ensemble

## 1 Introduction

Clustering, one of the most popular unsupervised learning methods, divides instances into clusters where instances in same cluster are similar while in different clusters are dissimilar [7]. However, algorithm selection and hyperparameter optimization are still two of the most challenging tasks for clustering problem.

In order to build high-quality clustering models, automated clustering as the subtask of Automated Machine Learning (AutoML) [20] has been proposed to address the above challenges. Existing automated clustering methods are widely

based on meta-learning [1, 3, 4, 6, 9, 13]. They learned from prior experience how different clustering models perform across datasets to speed up model design for given datasets [17, 20]. Despite the recent progress of meta-learning used in automated clustering, it still faces two specific problems: lacking comprehensive meta-features for meta-learning and general clustering validation index (CVI) as objective function in optimization process.

Meta-features play an important role in selecting promising algorithms or configurations in meta-learning based automated clustering. Most of the existing meta-features are extracted from labeled data, while applicable meta-features proposed for automated clustering are still incomprehensive. Reference [3] first studied meta-learning in clustering algorithm selection but they only used statistical meta-features. Later, meta-features from instances distance, link constraints, internal measures, and correlation for clustering datasets are proposed respectively [1, 4, 13, 19]. However, data distribution and the learning schema of clustering model are also important to characterize clustering datasets in meta-learning, which are related to the intrinsic features of clustering datasets.

Moreover, clustering validation indexes (CVIs) are used to measure the quality of clustering results. CVIs with different measuring criteria are suitable for specific clustering datasets and algorithms. Therefore, no general CVI is consistently superior to others in clustering validation [2, 11], which is one of the biggest challenges for model optimization in automated clustering. Reference [4] and [13] combined multiple CVIs and ranked algorithms based on their performance to choose the appropriate one. However, they are not robust since the error selection under any CVI can affect the overall algorithm ranking. In addition, the use of internal CVIs remains uncertain in hyperparameter optimization process for clustering evaluation. Hence, these methods still can not alleviate the dilemma of lacking general CVI.

In this paper, we propose a novel meta-learning based automated clustering method named AutoCluster to address the above problems, which is composed of Clustering-oriented Meta-feature Extraction (CME) and Multi-CVIs Clustering Ensemble Construction (MC<sup>2</sup>EC). The contributions of our work are highlighted as follows.

- In order to provide a more comprehensive characterization of clustering datasets, we propose CME for meta-learning. It extracts clustering-oriented meta-features from spatial randomness of data distribution and landmarker, i.e. running simple landmark clustering algorithms to fleetly capture the learning scheme, to enhance meta-learning.
- In order to alleviate the dilemma of lacking general CVI, we propose MC<sup>2</sup>EC for clustering model construction. It optimizes hyperparameters of promising algorithms suggesting by meta-learning under different CVIs and combines them to construct an ensemble model. Therefore, it provides a collaborative mechanism to balance the measuring criteria of different CVIs for discovering better clusters.
- In order to effectively build an appropriate clustering model for given datasets, we incorporate CME with MC<sup>2</sup>EC, and propose AutoCluster. It determines

promising clustering algorithms through CME-enhanced meta-learning under multiple CVIs, and performs automated ensemble construction based on these algorithms through MC<sup>2</sup>EC to provide appropriate clustering model.

- Finally, extensive experiments are conducted with a wide range of datasets from OpenML [18] and clustering benchmarks [5, 16], as well as various clustering algorithms from scikit-learn [12]. The results show the superiority of AutoCluster for building appropriate clustering model compared with classical clustering algorithms and CASH method.

The remainder of this paper is as follows: Sect. 2 presents the proposed automated clustering: AutoCluster. The extensive experiments for AutoCluster are analyzed in Sect. 3. Finally, we conclude this work in Sect. 4.

## 2 AutoCluster: Toward Automated Unsupervised Clustering

### 2.1 The Goal and Process of AutoCluster

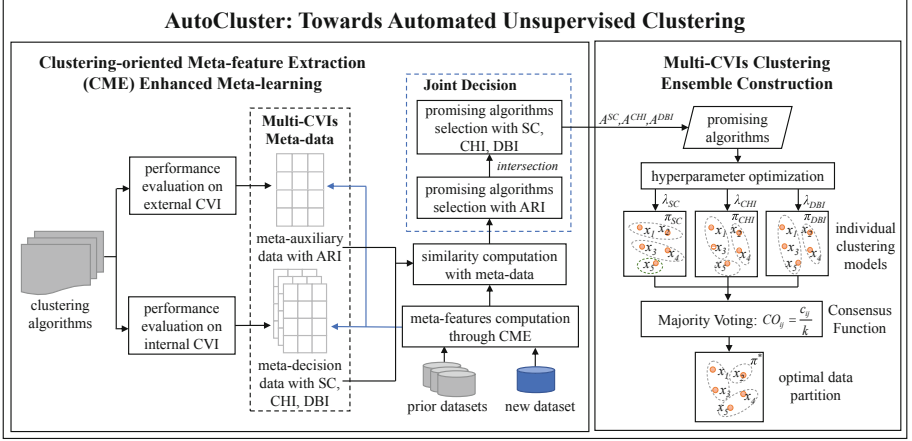
Automated clustering automatically builds appropriate clustering model for given datasets. In the process, AutoCluster has two specific problems: i) What meta-features can characterize unlabeled clustering datasets? ii) How to measure clustering result impartially? Thus, the goal of AutoCluster can be defined as follows.

**Definition 1 (AutoCluster).** For  $i = 1, 2, \dots, d$ , let  $x_i$  denote a feature vector of instance  $i$  without target value from clustering dataset  $D$ . Given a set of clustering algorithms  $A = \{A^1, A^2, \dots, A^m\}$ , and let the hyperparameters of each clustering algorithm  $A^i$  have domain  $\theta^i$ . The goal of AutoCluster is to discover more reasonable clusters  $\pi^*$  as Eq. 1.

$$\pi^* = \arg \min_{\pi \in Comb(A^*, \lambda^*)} C \left( \left\{ \arg \min_{A^i \in A, \lambda^i \in \theta^i} CVI_j(A^i, \lambda^i, D) \mid j = 1, \dots, c \right\} \right) \quad (1)$$

where  $CVI_j(A^i, \lambda^i, D)$  denotes CVI measured by clustering algorithm  $A^i$  with hyperparameter  $\lambda^i$  on dataset  $D$ , and  $c$  is the number of CVI in AutoCluster. Since AutoCluster handles the dilemma of lacking general CVI for model optimization by clustering ensemble,  $C(\cdot)$  represents consensus function to combine clustering model  $A^*$  with hyperparameter  $\lambda^*$  optimized with individual CVI.

As shown in Fig. 1, it is mainly composed of Clustering-oriented Meta-feature Extraction (CME) enhanced meta-learning and Multi-CVIs Clustering Ensemble Construction (MC<sup>2</sup>EC). For CME, traditional and clustering-oriented meta-features are extracted from data distribution and landmarker. The performance data with multiple CVIs is collected for meta-decision data and meta-auxiliary data. For MC<sup>2</sup>EC, it optimizes hyperparameters of promising algorithms suggesting from CME-enhanced-meta-learning under CVI metrics respectively through grid search and combines these clustering results to construct



**Fig. 1.** The process of AutoCluster

ensemble model through Majority Voting. Multiple CVIs and hyperparameter optimization process ensure AutoCluster can obtain diverse individual models with high quality for clustering ensemble to discover better data partition.

## 2.2 Enhanced Meta-learning for Finding Promising Algorithms

As the fundament of AutoCluster, meta-data is composed of meta-feature matrix and performance data. Here, we introduce the formulation of them respectively.

**Clustering-Oriented Meta-feature Extraction (CME) for Similarity Computation.** CME extracts five new clustering-oriented meta-features from data distribution and landmarker to provide a more comprehensive characterization for given datasets. It also extracts 19 traditional meta-features from [10]. The summary of meta-features is depicted in the supplementary material<sup>1</sup>.

The first meta-feature is from data distribution. Different clustering algorithms are suitable for the data with a specific distribution. Thus, data distribution is an important meta-feature in promising algorithm selection. Hopkins Statistic tests the spatial randomness of data distribution and also for cluster tendency which is defined as Eq. 2.

$$H = \frac{\sum_{i=1}^d u_i}{\sum_{i=1}^d u_i + \sum_{i=1}^d w_i} \quad (2)$$

where  $u_i$  represents the distance from  $d'$  sampling instances placed at random in the subspace of the entire  $h$ -dimensional sample space where  $d' \gg d$  to its nearest neighbor in dataset and  $w_i$  represents the distance from a randomly selected

<sup>1</sup> The supplementary material of this paper is available at <https://github.com/wj-tian/AutoCluster>.

instance to its nearest neighbor. For example, the hopkins statistic of regularly spaced and clustered datasets are always around 0.01 to 0.3 and 0.7 to 0.99 respectively. Thus, it can be an important meta-feature to describe datasets. Moreover, the learning properties of landmark clustering algorithms reflect the relative performance on given datasets, which are captured by learning scheme of landmark clustering algorithms. Specially, three different clustering algorithms are applied to extract landmarker meta-features: 1) The distance of the instances to their closest cluster center through KMeans (partition-based), which measures the compactness of cluster. 2) The number of leaves in the hierarchical tree through Agglomerative Clustering (hierarchy-based). 3) The reachability distances of instances and distance at which each instance becomes a core point through OPTICS (density-based), which measures density around an instance.

We denote  $F_i = \{f_1, f_2, \dots, f_{24}\}$  to be a feature vector of the enhanced meta-features of dataset  $D_i$ , where  $\{f_1, \dots, f_{19}\}$  are the traditional meta-features from [10] and  $\{f_{20}, \dots, f_{24}\}$  are clustering-oriented meta-features proposed by us. The distance metric between datasets determines how to find promising algorithms or configurations from the nearest dataset. In AutoCluster, we define p-norm distance in meta-features space to measure the similarity of datasets, which is computed as Eq. 3.

$$d_F = \|F_i - F_j\|_p \quad (3)$$

**Multi-performance Data for Promising Algorithms Selection.** Since AutoCluster develops a collaborative mechanism based on clustering ensemble to address the problem of lacking general CVI, every entry in performance data records the performance measured by different CVIs. We employ three internal CVIs, including two center-based representatives, Calinski-Harabasz Index (CHI) and Davies-Bouldin Index (DBI), and one non-center-based representative, Silhouette Coefficient (SC). They measure intra-cluster compactness and inter-cluster separation of a cluster from different criteria. Meanwhile, the prior datasets in meta-data have ground true labels. Thus, auxiliary information is extracted from external CVI. We apply Adjusted Rand index (ARI) to create meta-auxiliary data. The promising algorithm selection by ARI provides auxiliary information for internal CVIs as performing intersection to conduct joint decision and provide more promising algorithms.

### 2.3 Multi-CVIs Clustering Ensemble for Model Construction

No general CVI as objective function to measure clustering model impartially is one of the biggest challenges for model optimization in automated clustering. The important improvement of AutoCluster is to employ clustering ensemble to address this problem. In order to obtain better ensemble model, it requires diverse (making uncorrelated errors) and high-quality individuals for combination [15]. The application of multiple CVIs ensures the diverse generation of individuals. For high-quality individuals, MC<sup>2</sup>EC adopts grid search for a better configuration of promising algorithms. Suppose that a collection of optimized

models through grid search are obtained. The ensemble method apply in MC<sup>2</sup>EC is Majority Voting [8] to combine them, which is not conditioned by any particular clustering algorithm. It assumes that neighboring instances in ground true cluster are still likely to partition into same cluster, and then formulates consensus function based on co-association matrix to render pairs of instances voting for association in each partition produced by different clustering model. Each  $(i, j)$ th entry of instance  $x_i$  and  $x_j$  in co-association matrix is calculated as Eq. 4.

$$CO_{ij} = \frac{c_{ij}}{n_{SC} + n_{CHI} + n_{DBI}} \quad (4)$$

where  $n$  represents the number of optimized models with each CVI, and  $c_{ij}$  counts instance pair  $(x_i, x_j)$  co-occurring in same cluster. For final ensemble model, Majority Voting compares  $CO_{ij}$  in co-association matrix with a defined threshold  $\theta$ . Here, the final partition is formed with multiple CVIs.

### 3 Experiments

#### 3.1 Datasets and Clustering Algorithms

The evaluation of AutoCluster used 150 datasets in OpenML [18] sorted by most runs and selected by filtering with no more than 5000 samples and 50 features to create meta-data. Besides, 33 datasets are collected to test AutoCluster from Clustering basic benchmark [5]<sup>2</sup>, Fundamental clustering problem suite (FCPS) [16]<sup>3</sup> and Tomas Barton’s clustering benchmark<sup>4</sup>. More dataset information is described in Table 1. Six clustering algorithms are involved: KMeans, Affinity Propagation, Mean Shift, Agglomerative Clustering, DBSCAN, and Birch, which are implemented in scikit-learn [12] and corresponding hyperparameter spaces are depicted in the supplementary material.

**Table 1.** The summary of test datasets

No.	AutoCluster	Classes	Data points	Dimensions	No.	AutoCluster	Classes	Data points	Dimensions
1	a1	20	3000	2	18	Lsun	3	400	2
2	Aggregation	7	788	2	19	Lsun3D	4	404	3
3	aml28	5	804	2	20	Pathbased	3	300	2
4	Atom	2	800	3	21	R15	15	600	2
5	balance-scale	3	625	4	22	s1	15	5000	2
6	Compound	6	399	2	23	s2	15	5000	2
7	curves1	2	1000	2	24	s3	15	5000	2
8	curves2	2	1000	2	25	smile1	4	1000	2
9	D31	13	1232	2	26	Target	6	770	2
10	dietary_survey_IBS	2	400	42	27	Tetra	4	400	3
11	dim32	16	1024	32	28	unbalanced	8	6500	2
12	Flame	2	240	2	29	WingNut	2	1016	2
13	fourty	40	1000	2	30	zelnik1	3	299	2
14	gaussians1	2	100	2	31	zelnik5	4	512	2
15	Hepta	7	212	3	32	zelnik6	3	238	2
16	hypercube	8	800	3	33	zoo	7	101	16
17	Jain	2	373	2					

<sup>2</sup> <http://cs.uef.fi/sipu/datasets/>.

<sup>3</sup> <https://www.uni-marburg.de/fb12/arbeitsgruppen/datenbionik/data>.

<sup>4</sup> <https://github.com/deric/clustering-benchmark>.

### 3.2 Experimental Setup

All datasets are preprocessed by removing missing values, one-hot encoding for categorical features, and z-score standardization for all features. Our experiments selected three most similar datasets to perform majority selection of promising algorithms. 2-norm is used to compute the distance between datasets. The clustering ensemble is based on OpenEnsemble [14] and the threshold is set as 0.5. Moreover, our experiment is repeated 10 times to take the mean, and ARI is computed to measure the performance of AutoCluster to build clustering model. Then, the distance between hyperparameter configurations (HCD) is used to measure the diversity of individuals as defined in Eq. 5.

$$HCD = \frac{2}{N(N-1)} \sum_{i=1}^N \sum_{j=i+1}^N d^\lambda(\lambda^i, \lambda^j) \quad (5)$$

where  $N = n_{SC} + n_{CHI} + n_{DBI}$ , and  $d^\lambda(\lambda^i, \lambda^j)$  is equal to 1 when the algorithms of hyperparameter configuration  $\lambda^i$  and  $\lambda^j$  are different while the ratio of different hyperparameter values when  $\lambda^i$  and  $\lambda^j$  are for the same algorithm.

All procedures are run on Linux operating system with Intel(R) Xeon(R) Gold 6130 CPU @ 2.10 GHz processor. The process of running on a specific dataset is limited to a single CPU core.

### 3.3 Experimental Results and Analysis

**The Performance Evaluation of AutoCluster.** When tackling a specific problem by unsupervised clustering, many users lack enough experience to choose right algorithm or hyperparameter. It leads them to tend to choose algorithms with high reputations such as KMeans, and leave hyperparameter as default value or tuning the number of clusters with trial and error. In this basic experiment, we compare with six clustering algorithms with default hyperparameter values (KM-d, AP-d, MS-d, AC-d, DB-d, BI-d), and three KMeans algorithms with the number of clusters from 2 to 20 under SC, CHI, DBI respectively (SC-K, CHI-K, DBI-K). The result shown in Table 2 can be observed that AutoCluster obtains the highest ARI on 15/33 test datasets (the other test datasets are also close to the best methods), and the average ARI (0.776) dramatically surpasses the compared methods. Moreover, AutoCluster has a more stable prediction on these datasets since other compared methods only can perform well on few datasets, and these compared methods have more test datasets performing significantly worse. Thus, AutoCluster on different categories of datasets to automatically discover appropriate clusters is effective.

**The Performance Comparison of AutoCluster with CASH.** In this experiment, we compare with the most classic method named CASH to verify the superior of AutoCluster, in which the clustering algorithm selection is viewed as a super-hyperparameter and executed with hyperparameter optimization simultaneously. The objective function respectively sets as internal CVI

**Table 2.** The comparison with default hyperparameter values and simple optimization. For these compared methods, if the performance is highest, the corresponding entries are bolded, and if the performance is significantly worse than the highest performance on this dataset (lower than 0.3), the corresponding entries are underlined.

No.	AutoCluster	KM-d	AP-d	MS-d	AC-d	DB-d	BI-d	SC-K	CHI-K	DBI-K
1	0.904	<u>0.424</u>	<u>0.558</u>	<u>0.091</u>	<u>0.092</u>	<u>0.000</u>	<u>0.174</u>	<b>0.936</b>	0.930	0.835
2	<b>0.991</b>	0.664	<u>0.391</u>	<u>0.628</u>	<u>0.377</u>	<u>0.000</u>	<u>0.396</u>	0.762	<u>0.377</u>	0.762
3	0.996	<u>0.333</u>	<u>0.043</u>	0.975	0.859	0.999	0.959	0.975	<u>0.396</u>	<b>1.000</b>
4	0.528	<b>0.576</b>	0.519	0.548	<u>0.067</u>	0.568	<u>0.149</u>	0.547	0.537	0.535
5	0.121	<b>0.136</b>	0.034	0.000	0.121	0.000	0.111	0.127	0.095	0.104
6	<b>0.745</b>	0.456	<u>0.319</u>	0.722	0.484	0.740	0.734	0.721	0.721	0.721
7	<b>1.000</b>	<u>0.249</u>	<u>0.010</u>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.777	<b>1.000</b>	<u>0.099</u>	<b>1.000</b>
8	<b>0.523</b>	0.249	<u>0.061</u>	<u>0.179</u>	<u>0.019</u>	<u>0.000</u>	<u>0.130</u>	<u>0.199</u>	<u>0.098</u>	<u>0.098</u>
9	0.704	<u>0.670</u>	0.736	<u>0.254</u>	<u>0.126</u>	<u>0.319</u>	<u>0.274</u>	0.885	<b>0.976</b>	0.885
10	<b>1.000</b>	<u>0.360</u>	<u>0.137</u>	<u>0.689</u>	<b>1.000</b>	<u>0.000</u>	0.784	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
11	<b>1.000</b>	0.514	<u>0.363</u>	0.000	<u>0.123</u>	0.883	<u>0.175</u>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
12	<b>0.635</b>	<u>0.205</u>	<u>0.128</u>	0.524	<u>0.289</u>	<u>0.013</u>	<u>0.278</u>	0.425	<u>0.204</u>	0.426
13	<b>0.771</b>	<u>0.277</u>	0.614	<u>0.000</u>	<u>0.045</u>	<u>0.000</u>	<u>0.078</u>	0.636	0.628	0.625
14	<b>1.000</b>	<u>0.262</u>	<u>0.498</u>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
15	<b>1.000</b>	0.958	0.720	<u>0.000</u>	<u>0.269</u>	<b>1.000</b>	<u>0.354</u>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
16	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<u>0.000</u>	<u>0.222</u>	<b>1.000</b>	<u>0.426</u>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
17	<b>0.893</b>	<u>0.167</u>	<u>0.105</u>	<u>0.000</u>	<u>0.569</u>	<u>0.000</u>	<u>0.531</u>	<u>0.553</u>	<u>0.075</u>	<u>0.300</u>
18	0.528	0.390	<u>0.239</u>	0.420	<u>0.282</u>	<u>0.000</u>	0.559	<b>0.583</b>	0.326	<b>0.583</b>
19	0.864	<u>0.449</u>	<u>0.293</u>	<b>0.881</b>	0.602	<u>0.532</u>	0.788	0.599	0.599	0.599
20	<b>0.614</b>	0.404	<u>0.235</u>	<u>0.063</u>	0.414	<u>0.000</u>	0.468	0.480	<u>0.200</u>	<u>0.205</u>
21	0.989	<u>0.264</u>	<u>0.693</u>	<u>0.000</u>	<u>0.041</u>	<u>0.264</u>	<u>0.099</u>	<b>0.993</b>	<b>0.993</b>	<b>0.993</b>
22	<u>0.330</u>	0.500	<u>0.388</u>	<u>0.182</u>	<u>0.133</u>	<u>0.000</u>	<u>0.189</u>	0.551	<b>0.739</b>	0.552
23	0.916	<u>0.585</u>	<u>0.203</u>	<u>0.113</u>	<u>0.121</u>	<u>0.000</u>	<u>0.232</u>	0.938	0.938	<b>0.938</b>
24	0.685	0.481	<u>0.172</u>	<u>0.098</u>	<u>0.110</u>	<u>0.000</u>	<u>0.217</u>	<b>0.727</b>	0.727	0.727
25	0.749	<u>0.659</u>	<u>0.538</u>	<u>0.331</u>	<u>0.326</u>	<b>1.000</b>	<u>0.290</u>	0.707	<u>0.674</u>	<u>0.681</u>
26	<u>0.691</u>	<u>0.612</u>	<u>0.462</u>	<u>0.628</u>	<u>0.252</u>	<b>1.000</b>	<u>0.259</u>	<u>0.563</u>	<u>0.178</u>	<u>0.299</u>
27	<b>1.000</b>	<u>0.607</u>	<u>0.295</u>	<u>0.000</u>	<u>0.332</u>	<u>0.687</u>	0.713	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>
28	0.998	<b>1.000</b>	<u>0.193</u>	<u>0.612</u>	<u>0.124</u>	<u>0.610</u>	<u>0.127</u>	0.998	<u>0.450</u>	0.998
29	<b>1.000</b>	<u>0.226</u>	<u>0.073</u>	<u>0.000</u>	<b>1.000</b>	<u>0.000</u>	<u>0.476</u>	<u>0.670</u>	<u>0.264</u>	<u>0.156</u>
30	<u>0.308</u>	<u>0.280</u>	<u>0.266</u>	<u>0.000</u>	<u>0.024</u>	<b>0.630</b>	<u>0.112</u>	<u>0.288</u>	<u>0.260</u>	<u>0.269</u>
31	0.773	0.721	<u>0.360</u>	<u>0.000</u>	<u>0.310</u>	<b>1.000</b>	<u>0.226</u>	<u>0.603</u>	<u>0.301</u>	<u>0.437</u>
32	0.768	0.822	0.615	<b>0.829</b>	0.580	0.813	0.675	0.779	<u>0.323</u>	0.779
33	0.586	0.691	<u>0.480</u>	<u>0.034</u>	<u>0.448</u>	<u>-0.052</u>	0.677	<u>0.355</u>	<b>0.818</b>	<u>0.356</u>
Ave	<b>0.776</b>	0.491	0.356	0.327	0.356	0.424	0.407	0.715	0.574	0.663



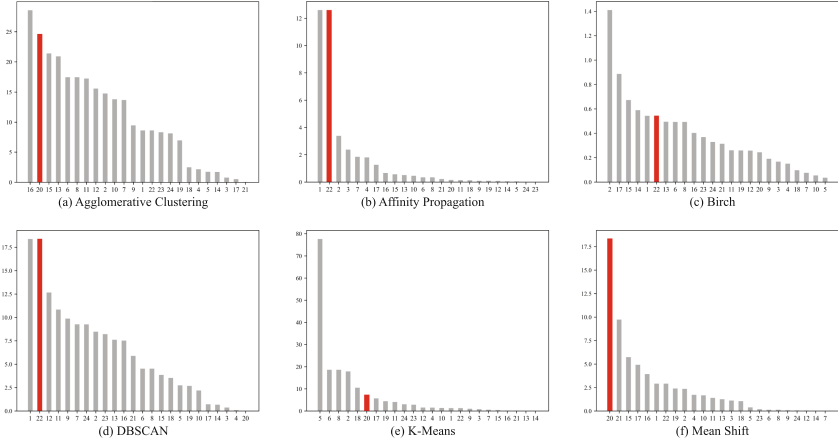
in AutoCluster (CASH-SC, CASH-CHI, and CASH-DBI). The result shown in Table 3 can be observed that AutoCluster performs better than CASH methods, which obtains the highest average ARI (0.776) and highest performance on 21/33 datasets. Moreover, AutoCluster is more stable over this experiment with one worse performance. CASH-SC, CASH-CHI, and CASH-DBI are uncomparable where the number of bolded/underlined entries is 12/5, 12/13, and 4/15 respectively. Thus, it can conclude that it is infeasible to directly introduce AutoML methods in supervised learning to unsupervised clustering problems to discover optimal partition.

**Why Does AutoCluster Work Well?** AutoCluster performs the effectiveness and superiority of discovering appropriate clusters for users automatically in comparison with classical clustering algorithms and CASH method. In this section, ablation studies of AutoCluster are conducted to interpret why AutoCluster works well and whether its components are reasonable.

*The Importance of CME.* In order to discover which meta-features are more important to algorithm selection, F-test is employed to evaluate the significant influence of meta-features on each algorithm. For each CVI and each algorithm, two groups of samples in meta-data are divided according to whether this algorithm is selected as a promising one under this CVI. Then, we calculate F-statistic between these two groups, which reflects whether the meta-features of any algorithm selected differ significantly from those of the algorithm not selected. We show the result of CHI as Fig. 2. From this figure, CME has a significant influence on algorithm selection, such as Agglomerative Clustering (No. 20 (*Hopkins Statistic*)), DBSCAN (No. 22 (*Agglomerative Clustering*)) and Mean Shift (No. 21 (*KMeans*)), which illustrates the importance of CME in AutoCluster. The remaining experimental results are described in the supplementary material.

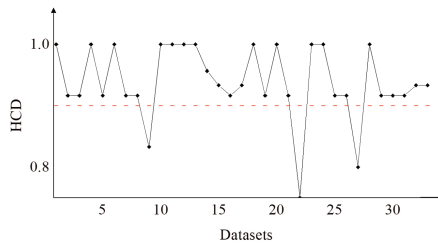
**Table 3.** The comparison with three CASH methods

No.	AutoCluster	CASH-SC	CASH-CHI	CASH-DBI	No.	AutoCluster	CASH-SC	CASH-CHI	CASH-DBI
1	0.904	<b>0.936</b>	0.916	<u>0.424</u>	18	0.528	<b>0.584</b>	0.319	0.582
2	<b>0.991</b>	0.776	<u>0.259</u>	0.771	19	<b>0.864</b>	0.597	0.599	0.591
3	<b>0.996</b>	0.975	<u>0.104</u>	0.988	20	<b>0.614</b>	0.476	<u>0.143</u>	0.539
4	0.528	<b>0.540</b>	0.244	0.518	21	0.989	<b>0.993</b>	<b>0.993</b>	0.851
5	0.121	<b>0.166</b>	0.082	0.022	22	<u>0.330</u>	0.548	<b>0.757</b>	0.595
6	0.745	0.721	0.721	<b>0.768</b>	23	0.916	0.938	<b>0.938</b>	<u>0.224</u>
7	<b>1.000</b>	<b>1.000</b>	<u>0.068</u>	<u>0.000</u>	24	0.685	0.711	<b>0.726</b>	<u>0.110</u>
8	<b>0.523</b>	<u>0.199</u>	<u>0.068</u>	<u>0.000</u>	25	<b>0.749</b>	0.733	0.649	<u>0.411</u>
9	0.704	0.855	<b>0.976</b>	0.792	26	<b>0.691</b>	<u>0.242</u>	<u>0.105</u>	<u>0.049</u>
10	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<u>0.501</u>	27	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.667
11	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<u>0.464</u>	28	<b>0.998</b>	0.998	0.287	<u>0.353</u>
12	<b>0.635</b>	0.425	<u>0.203</u>	<u>0.009</u>	29	<b>1.000</b>	0.693	<u>0.081</u>	<u>-0.001</u>
13	0.771	0.806	<b>0.845</b>	0.748	30	<b>0.308</b>	0.225	0.201	0.238
14	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	31	<b>0.773</b>	0.617	0.217	0.233
15	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	0.901	32	0.768	0.779	<u>0.222</u>	<b>0.858</b>
16	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	<b>1.000</b>	33	<b>0.586</b>	0.177	0.129	0.266
17	<b>0.893</b>	<u>0.553</u>	0.053	0.877	Ave	<b>0.776</b>	0.705	0.512	0.495

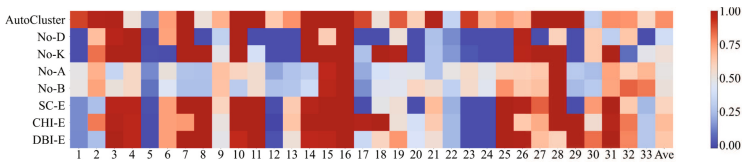


**Fig. 2.** The F-statics of meta-features grouped by the selected algorithm under CHI metric. No. 1–19 are for traditional meta-features from [10]. No. 20 is for the meta-feature of hopkins statistic and No. 21–24 are for landmarker meta-features

*The Effectiveness of Clustering Ensemble.* Clustering ensemble requires diverse and high-quality individual clustering models to construct ensemble model. Multiple CVIs and meta-learning with grid search ensure these two conditions in AutoCluster. HCD evaluates the diversity of hyperparameter configuration of individuals. The result shown in Fig. 3 illustrates the scores of HCD in clustering ensemble of all test datasets are higher than 0.75 and 31/33 datasets are higher than 0.9. Therefore, the high diversity of individuals in clustering ensemble makes AutoCluster effective to perform better.



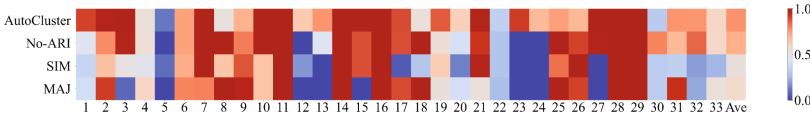
**Fig. 3.** The HCD of clustering ensemble on test datasets



**Fig. 4.** The comparison of disabling meta-learning and multiple CVIs

*The Comparison of Disabling Meta-learning and Multiple CVIs.* The key components of AutoCluster are CME-enhanced meta-learning and MC<sup>2</sup>EC. Thus, we compare with the following methods: 1) No-D, No-K, No-A, No-B: They disable meta-learning. No-D is heterogeneous ensemble with six clustering algorithms with default hyperparameters involved in AutoCluster. No-K, No-A, and No-B are isomorphic clustering ensemble of KMeans, Agglomerative Clustering, and Birch with different numbers of clusters. 2) SC-E, CHI-E, DBI-E: These methods disable multiple CVIs. They execute meta-learning and clustering ensemble under a single CVI. As shown in Fig. 4, AutoCluster achieves drastic improvement with meta-learning and multiple CVIs. It has the highest performance on 18/33 test datasets, and the best average ARI of these two categories of methods are 0.532 and 0.675 respectively, which are well below 0.776 of AutoCluster. Since the methods disabling meta-learning lead to the individuals with low quality and the methods disabling multiple CVIs lead to the individuals with low diversity, they both fail to discover promising partition. Therefore, the result can show the necessity of meta-learning and multiple CVIs in AutoCluster.

*The Comparison of Selection Strategy of Individuals in Clustering Ensemble.* The evaluation of the individual selection based on ARI in AutoCluster is compared with three methods: 1) No-ARI: It directly executes clustering ensemble from meta-learning under internal CVIs. 2) SIM: It selects individuals based on similarity of clustering result measuring by Normalized Mutual Information. When it is greater than 0.8, one of these two models is removed from ensemble. 3) MAJ: When the algorithms are selected by majority (2/3) CVIs, their corresponding models are selected for final ensemble.



**Fig. 5.** The comparison of selection strategy of individual models

The result is shown in Fig. 5. Since the measure criteria of internal CVIs are difficult to fit the ground true label, it is important for meta-learning to provide external auxiliary information for clustering ensemble. No-ARI directs to select individuals without auxiliary information, where it only performs best on 11 test datasets, worse than AutoCluster of 23, and the average ARI is 0.131 lower than AutoCluster. SIM inevitably removes promising individuals when the high similarity between them. Thus, it fails to surpass AutoCluster where the number of best datasets is 7 and the average ARI is 0.526. MAJ method also can remove promising individuals since different criteria of CVIs. It leads the bad performance compared with AutoCluster with 12 best datasets and the average ARI of 0.567. Therefore, it shows the feasibility of individual selection to provide external auxiliary information in AutoCluster. In addition, the runtime

of AutoCluster on 33 test datasets is also considered to verify the efficiency of this method, which is depicted in the supplementary material.

## 4 Conclusions

Automated clustering based on meta-learning faces its specific problems: lacking comprehensive meta-features and general CVI. This paper proposes a novel automated clustering method named AutoCluster, mainly composed of CME and MC<sup>2</sup>EC. CME extracts five clustering-oriented meta-features to extend traditional meta-features from spatial randomness and learning properties of clustering algorithms. MC<sup>2</sup>EC develops a collaborative mechanism to balance the measuring criterion of different CVIs based on clustering ensemble. Extensive experiments are conducted with a wide range of datasets from OpenML and three- clustering benchmarks. The results show that AutoCluster has strong ability to construct appropriate clustering model than compared methods.

Meta-learning and clustering ensemble are important to promote the performance of AutoCluster. Hence, applying meta-features into manifold space or importance-weighted space is promising in future works. For clustering ensemble, optimized clustering ensemble methods like evidence accumulation also can be applied to improve the performance of AutoCluster.

**Acknowledgment.** This work is supported by the National Natural Science Foundation of China (No. 52073169) and the State Key Program of National Nature Science Foundation of China (Grant No. 61936001).

## References

1. Adam, A., Blockeel, H.: Dealing with overlapping clustering: a constraint-based approach to algorithm selection. In: *Meta-Learning and Algorithm Selection workshop-ECMLPKDD2015*, vol. 1, pp. 43–54 (2015)
2. Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M., Perona, I.: An extensive comparative study of cluster validity indices. *Pattern Recogn.* **46**(1), 243–256 (2013)
3. De Souto, M.C., et al.: Ranking and selecting clustering algorithms using a meta-learning approach. In: *2008 IEEE International Joint Conference on Neural Networks*, pp. 3729–3735 (2008)
4. Ferrari, D.G., De Castro, L.N.: Clustering algorithm selection by meta-learning systems: a new distance-based problem characterization and ranking combination methods. *Inf. Sci.* **301**, 181–194 (2015)
5. Fränti, P., Sieranoja, S.: K-means properties on six clustering benchmark datasets. *Appl. Intell.* **48**(12), 4743–4759 (2018)
6. Garg, V., Kalai, A.T.: Supervising unsupervised learning. *Adv. Neural Inf. Process. Syst.* **31**, 4991–5001 (2018)
7. Jain, A.K.: Data clustering: 50 years beyond k-means. *Pattern Recogn. Lett.* **31**(8), 651–666 (2010)
8. Jamali, N., Sammut, C.: Majority voting: material classification by tactile sensing using surface texture. *IEEE Trans. Robot.* **27**(3), 508–521 (2011)

9. José-García, A., Gómez-Flores, W.: Automatic clustering using nature-inspired metaheuristics: a survey. *Appl. Soft Comput.* **41**, 192–213 (2016)
10. Li, Y.F., Wang, H., Wei, T., Tu, W.W.: Towards automated semi-supervised learning. In: *AAAI*, vol. 33, pp. 4237–4244 (2019)
11. Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J.: Understanding of internal clustering validation measures. In: *ICDM*, pp. 911–916 (2010)
12. Pedregosa, F., et al.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
13. Pimentel, B.A., de Carvalho, A.C.: A new data characterization for selecting clustering algorithms using meta-learning. *Inf. Sci.* **477**, 203–219 (2019)
14. Ronan, T., Anastasio, S., Qi, Z., Sloutsky, R., Naegle, K.M., Tavares, P.H.S.V.: Openensembles: a python resource for ensemble clustering. *J. Mach. Learn. Res.* **19**(1), 956–961 (2018)
15. Topchy, A., Jain, A.K., Punch, W.: Combining multiple weak clusterings. In: *Proceedings of the Third IEEE International Conference on Data Mining*, pp. 331–338 (2003)
16. Ultsch, A.: Clustering with som: U\* c. In: *Proceedings of the Workshop on Self-Organizing Maps, 2005* (2005)
17. Vanschoren, J.: Meta-learning: a survey. *CoRR* abs/1810.03548 (2018)
18. Vanschoren, J., Van Rijn, J.N., Bischl, B., Torgo, L.: OpenML: networked science in machine learning. *ACM SIGKDD Explor. Newsl.* **15**(2), 49–60 (2014)
19. Vukicevic, M., Radovanovic, S., Delibašić, B., Suknovic, M.: Extending meta-learning framework for clustering gene expression data with component based algorithm design and internal evaluation measures. *Int. J. Data Min. Bioinform.* **14**, 101–119 (2016)
20. Zöller, M., Huber, M.F.: Benchmark and survey of automated machine learning frameworks. *J. Artif. Intell. Res.* **70**, 409–472 (2021)