J. Yaghob (Ed.): ITAT 2015 pp. 108–114
Charles University in Prague, Prague, 2015

Using Multi-Objective Optimization for the Selection of Ensemble Members

Tomáš Bartoň[1,2] Pavel Kordík[1]

[1] Faculty of Information Technology
Czech Technical University in Prague, Thakurova 9, Prague 6, Czech Republic
http://www.fit.cvut.cz
tomas.barton@fit.cvut.cz
[2] Institute of Molecular Genetics of the ASCR, v. v. i.
Vídeňská 1083, 142 20 Prague 4, Czech Republic

Abstract: In this paper we propose a clustering process which uses a multi-objective evolution to select a set of diverse clusterings. The selected clusterings are then combined using a consensus method. This approach is compared to a clustering process where no selection is applied. We show that careful selection of input ensemble members can improve the overall quality of the final clustering. Our algorithm provides more stable clustering results and in many cases overcomes the limitations of base algorithms.

1 Introduction

Clustering is a popular technique that can be used to reveal patterns in data or as a preprocessing step in data analysis. Clustering tends to group similar objects into groups that are called clusters and to place dissimilar objects into different clusters. Its application domains include data mining, information retrieval, bioinformatics, image processing and many others.

Many clustering algorithms have been introduced so far, an extensive overview of clustering techniques can be found in [1]. Since we have the No Free Lunch Theorem [36], which states that there is no single, supreme algorithm that would optimally run on all datasets, it is complicated for a user to decide which algorithm to choose.

Ensemble methods have been successfully applied in supervised learning [25] and similar attempts appeared in the unsupervised learning area [32], [14], [34]. Cluster ensemble methods should eliminate the drawbacks of individual methods by combining multiple solutions into one final clustering.

2 Cluster Ensembles

The main objective of clustering ensembles is to combine multiple clusterings into one, preferably high-quality solution.

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n data points, where each $x_i \in X$ is represented by a vector of d attributes. A cluster ensemble is defined as $\Pi = \{\pi_1, \pi_2, \ldots, \pi_m\}$ with m clusterings. Each base clustering π_i consists of a set of clusters $\pi_i = \{C_1^i, C_2^i, \ldots, C_{k_i}^i\}$, such that $\cup_{j=1}^{k_i} C_j^i = X$, where k_i is the number of clusters in a given ensemble member (it does not have to be the same for all members).

The problem is how to obtain a final clustering $\pi^* = \{C_1^*, C_2^*, \ldots, C_K^*\}$, where K is the number of clusters in the result and π^* summarizes the information from ensemble Π.

The process consists of two major steps. Firstly, we need to generate a set of clustering solutions and secondly we need ti combine the information from these solutions. A typical result of the ensemble process is a single clustering. It has been shown in supervised ensembles that the best results are achieved when using a set of predictors whose errors are dissimilar [25]. Thus it is desirable to introduce diversity between ensemble members.

2.1 Ensemble Generation Strategies

Several approaches have been used to initialize clustering solutions in order to create an ensemble.

- **Homogeneous ensembles**: Base clusterings are created using repeated runs of a single clustering algorithm. This is quite a popular approach, repeated runs of k-means with random center initialization have been used in [14]. When using k-means the number of clusters is typically fixed to $\lceil \sqrt{n} \rceil$, where n is the size of the dataset [22].
- **Varying k**: Repeated runs of k-means with random initialization and k [15], a golden standard is using k in the range from 2 to $\sqrt{n}$.
- **Random subspacing** An ensemble is created from base clusterings that use different initial data. This could be achieved by projecting data onto different subspaces [13], [16] choosing different subsets of features [32], [37], or using data sampling techniques [10].
- **Heterogeneous ensembles**: Diversity of solutions is introduced by applying different algorithms on the same dataset.

2.2 Consensus Functions

Having multiple clusterings in an ensemble, many functions have been proposed to derive a final clustering. When only one solution is considered as the result, it is usually referred as a consensus function, unlike meta clustering where the output is a set of multiple clusterings [6].

There are several approaches as to how to represent information contained in base clusterings, some use matrices while others use graph representation.

- **Pairwise similarities**: A pairwise similarity matrix is created and afterwards a clustering algorithm (e.g. hierarchical agglomerative clustering) is applied to group together items that were most frequently together in the same cluster in all the base clusterings [14]. the Cluster-based Similarity Partitioning Algorithm (CSPA) from Strehl and Ghosh [32] uses METIS [24] for partitioning a similarity matrix into k components.

- **Feature-based approach**: The ensemble problem is formulated as categorical data clustering. For each data point an m-dimensional vector containing labels in base clusterings is created. The goal is to find a partition π^* which summarizes the information gathered from the partitions Π [28], [33], [4].

- **Graph based**: Many methods use graph representation for capturing relationships between base clusterings. Strehl and Ghosh [32] also proposed the HyperGraph-Partitioning Algorithm (HGPA), where vertices correspond to data points and a hyperedge represents clusters. Another approach chooses COMUSA [27] which increases the weight of the edge for each occurrence of data pairs in the same cluster. Afterwards the nodes are sorted by the attachment score, which is defined as the ratio between the sum of the node's weights and its number of incident edges. The nodes with the highest attachment score are then used as a foundation for new clusters. This approach is relatively fast to compute, however it might fail to capture complex relationships between very diverse clusterings.

3 Multi-Objective Clustering

Multi-objective clustering usually optimizes two objective functions. Using more than a few objectives is not usual because the whole process of optimization becomes less effective.

The first multi-objective evolutionary clustering algorithm was introduced in 2004 by Handl and Knowles [18] and is called VIENNA (the Voronoi Initialized Evolutionary Nearest-Neighbour Algorithm).

Subsequently, in 2007 Handl and Knowles published a Pareto-based multi-objective evolutionary clustering algorithm called MOCK [19] (Multi-Objective Clustering with automatic K-determination). Each individual in MOCK is represented as a directed graph which is then translated into a clustering. The genotype is encoded as an array of integers whose length is same as the number of instances in the dataset. Each number is a pointer to another instance (an edge in the graph), since it is connected to the instance at a given index. This easily enables the application of mutation and crossover operations.

As a Multi-Objective Evolutionary Algorithm (MOEA), MOCK employs the Pareto Envelope-based Selection Algorithm version 2 (PESA-II) [7], which keeps two populations, an internal population of fixed size and a larger external population which is exploited to explore good solutions. Two complementary objectives, *deviation* and *connectivity*, are used as objectives in the evolutionary process.

A clear disadvantage of MOCK is its computation complexity, which is a typical characteristic of evolutionary algorithms. Nevertheless, the computation time spent on MOCK should result in high-quality solutions. Faceli et al. [12] reported that for some high-dimensional data it is not guaranteed that the algorithm will complete, unless the control front distribution has been adjusted for the given data set.

Faceli et al. [12] combined a multi-objective approach to clustering with ensemble methods and the resulting algorithm is called MOCLE (Muli-Objective Clustering Ensemble Algorithm). The objectives used in the MOCLE algorithm are the same as those used in MOCK [19]: deviation and connectivity. Unlike MOCK, in this case the evolutionary algorithm used is NSGA-II [9].

4 Our Approach

There are many ways to produce a final consensus, nonetheless in this work we focus on the selection of high-quality and diverse clusterings.

In order to optimize ensemble member selection, we apply a multi-objective optimization. The whole process is shown in Fig. 1. In our previous work [3] we have shown that introducing multiple objectives into the clustering process can improve the quality of clustering, specifically using the Akaike information criterion (AIC) [2] (or BIC [31]) with another criterion leads to better results. The second criterion is typically based on computing a ratio between cluster compactness and the sum of distances between cluster centres.

AIC is typically used in supervised learning when trying to estimate model error. Essentially it attempts to estimate the optimism of the model and then add it to the error [20]:

$$f_{AIC} = -2 \cdot \frac{\log(L_n(k))}{n} + 2 \cdot \frac{k}{n} \tag{1}$$

where $L_n(k)$ is the maximum likelihood of a model
with k parameters based on a sample of size n.

The SD index was introduced in 2000 by Halkidi et al. [17] and it is based on concepts of *average cluster scattering* and *total separation of clusters* which were previously used by Rezaee et al. [29] for evaluation of fuzzy clusterings.

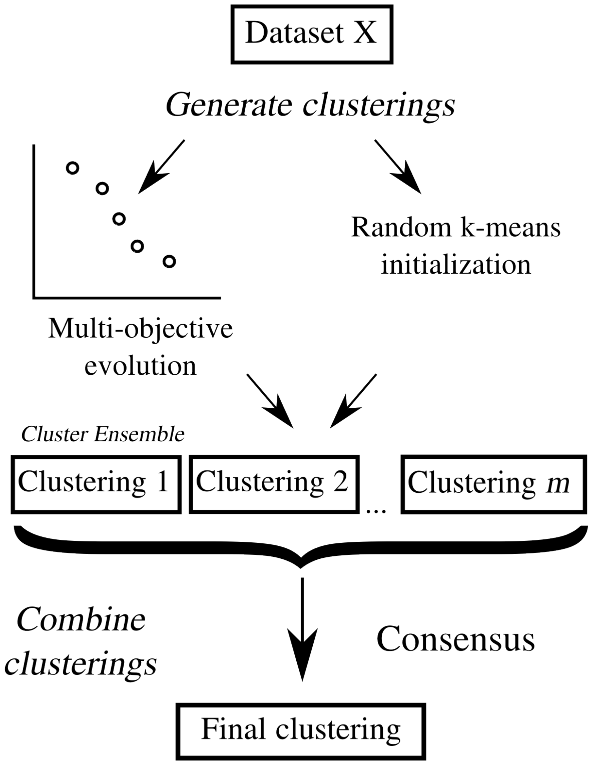

Figure 1: Cluster Ensemble Process: Firstly using the same input dataset we generate multiple clusterings, then select m best clusterings for the ensemble and combine them into a single solution.

The average scattering is defined as:

$$Scatt(k) = \frac{1}{k}\sum_{i=1}^{k}\frac{\|\sigma(\bar{\mathbf{c}}_i)\|}{\|\sigma(X)\|} \tag{2}$$

where $\|\mathbf{x}\|$ is the norm of a vector,
$\bar{\mathbf{c}}_i$ is the centroid of the i-th cluster,
$\sigma(X)$ is the variance of the input dataset.

$\sigma(X) \in \mathbb{R}^m$ with m being the number of dataset dimensions. Variance for a dimension d is defined as:

$$\sigma^d = \frac{1}{n}\sum_{i=1}^{n}\left(x_i^d - \bar{x}^d\right)^2$$

$$\|\sigma(X)\| = \sqrt{\sum_{i=1}^{m}(\sigma^d)^2}$$

The total separation is given by:

$$Dis(k) = \frac{D_{max}}{D_{min}}\sum_{i=1}^{k}\left(\sum_{j=1}^{k}\left\|\bar{\mathbf{c}}_i - \bar{\mathbf{c}}_j\right\|\right)^{-1} \tag{3}$$

where D_{max} is the maximum distance and D_{min} is the minimum distance between cluster centers ($\bar{c}_i$) and k is the number of clusters.

$$D_{max} = \max_{\substack{i,j\in\{1,\dots,k\}\\ i\neq j}}\left(\left\|\bar{\mathbf{c}}_i - \bar{\mathbf{c}}_j\right\|\right) \tag{4}$$

$$D_{min} = \min_{\substack{i,j\in\{1,\dots,k\}\\ i\neq j}}\left(\left\|\bar{\mathbf{c}}_i - \bar{\mathbf{c}}_j\right\|\right) \tag{5}$$

Then we can define the SD validity index as follows:

$$f_{SD} = \alpha \cdot Scat(k) + Dis(k) \tag{6}$$

where α should be a weighting factor equal to $Dis(c_{max})$ with c_{max} being the maximum number of clusters [17]. This makes perfect sense for fuzzy clustering (as was proposed in [29]), however it is rather unclear how to compute c_{max} in the case of crisp clustering, when $c_{max} \gg k$ without running another clustering with c_{max} as the requested number of clusters. Nonetheless, [17] mentions that "SD proposes an optimal number of clusters almost irrespective of c_{max}, the maximum number of clusters", thus we consider the special case where $c_{max} = k$:

$$\begin{aligned} f_{SD} &= Dis(k)\cdot Scat(k) + Dis(k) && (7)\\ &= Dis(k)\cdot(Scat(k)+1) && (8)\end{aligned}$$

The idea of minimizing inner cluster distances and maximizing distances between cluster centres it not really new. In fact most of the clustering objective functions use this idea. Instead of the SD index we could have used C-index [21], Calinski-Harabasz Index [5], Davies-Bouldin [8] or Dunn's index [11], just to name a few. Each of these indexes might perform better on some dataset, however it is out of scope of this paper to compare which combination would be better. We considered

As an evolutionary algorithm we use NSGA-II [9] while using only the mutation operator and a relatively small population (see Table 2). Each individual in the population contains a configuration of a clustering algorithm. For this experiment we used only the basic k-means [26] algorithm with random initialization. The number of clusters k is the only parameter that we change during the evolutionary process. For k we used a constraint in order to keep the number of clusters within reasonable boundaries. For all test datasets we used the interval $\langle 2, \sqrt{n}\rangle$.

In order to evaluate the effect of Pareto optimal solutions selection for the ensemble bagging process, we compare two strategies for generating base clusterings. The first one generates 10 k-means clusterings with random initialization with k within the interval $\langle 2, \sqrt{n}\rangle$. The second method uses multi-objective evolution of clusterings where each individual represents a k-means clustering.

As a consensus function we use either a graph based COMUSA algorithm or a hierarchical agglomerative clustering of the co-association matrix. Both approaches have their drawbacks, the COMUSA algorithm is locally consistent, however fails to integrate the information contained in different base clusterings, thus final clustering does not overcome limitations of base clustering algorithms. The latter could be very slow on larger datasets (complexity $O(n^3)$) and might produce noisy clustering in case of very diverse base clusterings.

Dataset	**d**	**n**	**classes**	**source**
aggregation	2	788	7	[15]
atom	3	800	2	[35]
chainlink	3	1000	2	[35]
complex8	2	2551	8	[30]
complex9	2	3031	9	[30]
diamond9	2	3000	9	[30]
jain	2	373	2	[23]
long1	2	1000	2	[18]
longsquare	2	900	6	[18]
lsun	2	400	3	[35]
target	2	770	6	[35]
tetra	3	400	4	[35]
triangle1	2	1000	4	[18]

Table 1: Datasets used for experiments.

Parameter	Setting
Number of generations	5
Population size	10
Mutation probability	30%
Mutation	Polynomial mutation

Table 2: Parameter settings for NSGA-II evolution

To evaluate clustering quality we use NMI[1] (Normalized Mutual Information) [32]. NMI has proved to be a better criterion than the Adjusted Rand Index (ARI) for the evaluation of datasets with apriori known labels. However it does not capture noise in clusters (which does not occur in the case of traditional *k*-means), which might be introduced by some consensus methods. Another limitation of these evaluation metrics is apparent from Figure 6, where clustering with obvious 50% assignment error gets lowest possible score. Both indexes do not penalize correctly higher number of clusters, thus algorithms producing many small clusters will be preferred by these objectives.

5 Results

Most of the datasets used in these experiments come from the Fundamental Clustering Problems Suite [35]. We have intentionally chosen low dimensional data in order to be able to visually evaluate the results. An overview of the datasets used can be found in Table 1.

As objectives during the multi-objective evolution we used AIC and SD index in all cases. The advantage is combination of two measures that are based on very different grounds. We tried using Deviation and Connectivity, as it was proposed by Handl [19] (and later used by [12]), but for all tested datasets, our approach was better or at least comparable with these objectives.

Dataset	**C-RAND**	**C-MO**	***k*-means**
aggregation	0.75	**0.84**	**0.84**
atom	0.39	**0.61**	0.28
chainlink	0.07	**0.50**	0.07
complex8	0.59	**0.64**	0.59
complex9	0.64	**0.66**	0.65
diamond9	**1.00**	0.87	0.97
jain	0.36	**0.51**	0.37
long1	0.00	**0.49**	0.03
longsquare	0.83	**0.86**	0.80
lsun	0.54	**0.69**	0.54
target	0.67	0.57	**0.69**
tetra	**1.00**	0.86	0.99
triangle1	**0.95**	0.78	0.94

Table 3: NMI_{sqrt} score from various artificial dataset clusterings. C-RAND uses bagging of 10 k-means runs with a fixed value *k* that corresponds to the number of classes in the given dataset. The same approach is used in the case of *k*-means, but without bagging. C-MO is a NSGA-II based algorithm with criteria AIC and SD index; the number of clusters is randomized during evolution within the interval 2 to $\sqrt{n}$. Both C-RAND and C-MO use the COMUSA algorithm to form final (consensus) partitioning. The NMI values are averages from 30 independent runs.

In the case of the COMUSA approach, the multi-objective selection either improved the final NMI score or it was comparable with the random selection. There were only a few exceptions, especially in case of compact datasets like the 9 diamond dataset which contains a specific regular pattern that is unlikely to appear in any real-world dataset (see Table 3). The number of clusters produces by COMUSA approach was usually higher than the correct number of clusters and clustering contains many nonsense clusters. Despite these facts the NMI score does not penalize these properties appropriately.

Agglomerative clustering of the co-association matrix despite its high computational requirement provide pretty good results. Nonetheless in the case of the *chainlink* dataset there is over 50% improvement in NMI. It is important to note, that for HAC-RAND and HAC-MO we did not provide the information about the correct number of clusters. Still these methods manages to estimate number of clusters or at least provide result that is very close to the number of supervised classes.

[1] There are several versions of this evaluation method, sometimes it is referred as NMI_{sqrt}

Dataset	HAC-RAND	HAC-MO
aggregation	**0.74**	0.71
atom	0.44	**0.68**
chainlink	0.47	**0.99**
complex8	**0.72**	0.71
complex9	**0.71**	0.70
diamond9	**0.83**	0.75
jain	**0.61**	0.60
long1	0.73	**0.81**
longsquare	0.73	**0.89**
lsun	0.66	**0.71**
target	0.40	0.40
tetra	0.75	**0.99**
triangle1	0.78	**0.89**

Table 4: NMI_{sqrt} score from the clustering of various artificial datasets. As a consensus method, hierarchical agglomerative clustering (with complete linkage) of co-association matrix is used. In the case of *HAC-RAND* we run 10 independent k-means clusterings with a random number of clusters (between 2 and $\sqrt{n}$), then form a co-association matrix and finally run agglomerative clustering of the matrix. *HAC-RAND* works in very similar manner, but instead of the first step a multi-objective evolution of k-means is performed. 10 dominating solutions are selected and the rest of the algorithm is the same.

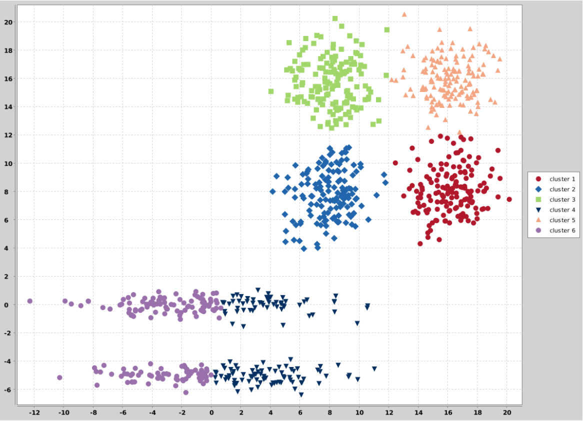

Figure 2: Typical clustering *longsquare* dataset using k-means ($k = 6$). K-Means algorithm fails to reveal non-spherical clusters (Ajusted Rand Index = 0.94, NMI = 0.85).

6 Conclusion

During our experiments we have shown that careful selection of clusterings for the ensemble process can significantly improve overall clustering quality for non-trivial datasets (measured by NMI).

It is interesting that non-spherical clusters could be discovered by consensus function when agglomerative hierarchical clustering is used (compare Fig. 2 and 3).

Using a multi-objective optimization for clustering selection improved the overall quality of clustering results, however ensemble methods might produce noisy clustering with a higher evaluation score. Noisy clusterings are hard to measure with current evaluation metrics, therefore it might be beneficial to include an unsupervised score in the results. In further research we would like to examine the process of selecting optimal objectives for each dataset.

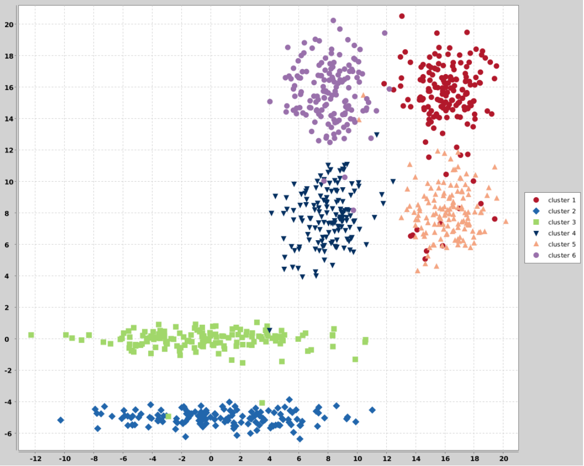

Figure 3: Consensus of 10 independent k-means runs on the *longsquare* dataset using $k = 6$. The co-association matrix obtained was clustered using hierarchical clustering (average linkage). The resulting clustering is qualitatively better than a single k-means run, however it contains noise in all clusters (Ajusted Rand Index = 0.99, NMI = 0.95).

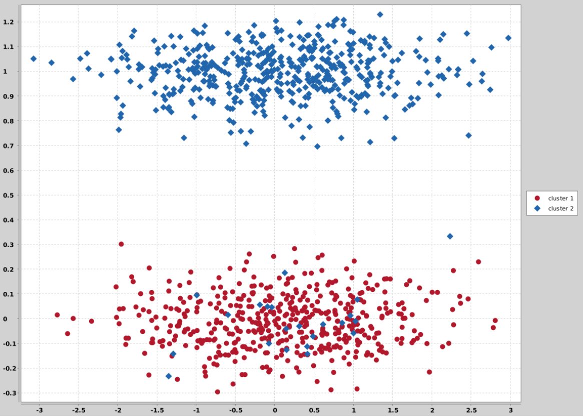

Figure 4: Consensus of 10 independent k-means runs on the *long1* dataset using $k = 6$. On many datasets setting the correct k does not improve the quality of the resulting clustering. The co-association matrix obtained was clustered using hierarchical clustering (average linkage).

Acknowledgement

The research reported in this paper was supported by the Czech Science Foundation (GAČR) grant 13-17187S.

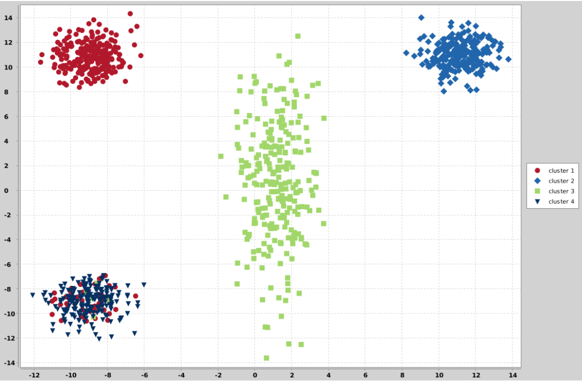

Figure 5: Consensus of 10 independent k-means runs on the *triangle1* dataset using $k = 6$. In this case the consensus produces a nonsense cluster that mixes items from 3 other clusters together.

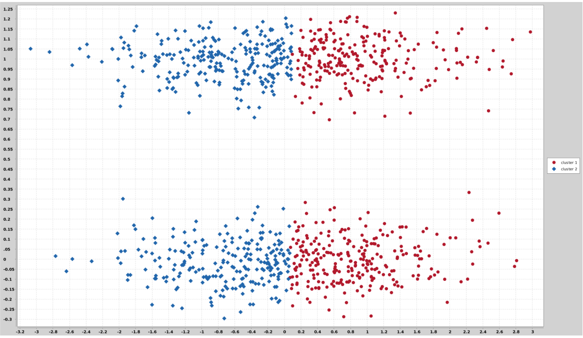

Figure 6: Single k-means runs on the *long1* dataset using $k = 2$. K-Means algorithm fails to reveal non-spherical clusters. Both supervised indexes Ajusted Rand Index and NMI assigns this clustering score 0.0, even though there is 50% assignment error.

References

[1] Aggarwal, C. C., Reddy, C. K., (Eds.): Data clustering: algorithms and applications. CRC Press, 2014

[2] Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control **19** (6 December 1974), 716–723

[3] Bartoň, T., Kordík, P.: Evaluation of relative indexes for multi-objective clustering. In: Hybrid Artificial Intelligent Systems, E. Onieva, I. Santos, E. Osaba, H. Quintián, and E. Corchado, (Eds.), vol. 9121 of Lecture Notes in Computer Science. Springer International Publishing, 2015, 465–476

[4] Boulis, C., Ostendorf, M.: Combining multiple clustering systems. In: PKDD (2004), J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, (Eds.), vol. 3202 of Lecture Notes in Computer Science, Springer, 63–74

[5] Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics-theory and Methods **3 (1)** (1974), 1–27

[6] Caruana, R., Elhawary, M., Nguyen, N., Smith, C.: Meta clustering. In: Proceedings of the Sixth International Conference on Data Mining (Washington, DC, USA, 2006), ICDM'06, IEEE Computer Society, 107–118

[7] Corne, D., Jerram, N., Knowles, J., Oates, M.: PESA-II: region-based selection in evolutionary multiobjective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001) (2001)

[8] Davies, D. L., Bouldin, D. W.: A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. **1 (2)** (1979), 224–227

[9] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6 (2)** (2002), 182–197

[10] Dudoit, S., Fridlyand, J.: Bagging to improve the accuracy of a clustering procedure. Bioinformatics **19 (9)** (2003), 1090–1099

[11] Dunn, J. C.: Well-separated clusters and optimal fuzzy partitions. Journal of Cybernetics **4 (1)** (1974), 95–104

[12] Faceli, K., de Souto, M. C. P., de Araujo, D. S. A., de Carvalho, A. C. P. L. F.: Multi-objective clustering ensemble for gene expression data analysis. Neurocomputing **72 (13-15)** (2009), 2763–2774

[13] Fern, X. Z., Brodley, C. E.: Random projection for high dimensional data clustering: a cluster ensemble approach. In: ICML (2003), T. Fawcett and N. Mishra, (Eds.), AAAI Press, 186–193

[14] Fred, A. L. N., Jain, A. K.: Combining multiple clusterings using evidence accumulation. IEEE Trans. Pattern Anal. Mach. Intell. **27 (6)** (2005), 835–850

[15] Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. TKDD **1 (1)** (2007)

[16] Gullo, F., Domeniconi, C., Tagarelli, A.: Projective clustering ensembles. Data Min. Knowl. Discov. **26 (3)** (2013), 452–511

[17] Halkidi, M., Vazirgiannis, M., Batistakis, Y.: Quality scheme assessment in the clustering process. In: PKDD (2000), D. A. Zighed, H. J. Komorowski, and J. M. Zytkow, (Eds.), vol. 1910 of Lecture Notes in Computer Science, Springer, 265–276

[18] Handl, J., Knowles, J.: Evolutionary multiobjective clustering. Parallel Problem Solving from Nature – PPSN VIII (2004), 1081–1091

[19] Handl, J., Knowles, J.: An evolutionary approach to multiobjective clustering. Evolutionary Computation, IEEE Transactions on **11 (1)** (2007), 56–76

[20] Hastie, T., Tibshirani, R., Friedman, J., Corporation, E.: The elements of statistical learning. Springer, Dordrecht, 2009

[21] Hubert, L., Levin, J.: A general statistical framework for assessing categorical clustering in free recall. Psychological Bulletin **83 (6)** (1976), 1072

[22] Iam-on, N., Boongoen, T.: Improved link-based cluster ensembles. In: IJCNN (2012), IEEE, 1–8.

[23] Jain, A. K., Law, M. H. C.: Data clustering: a user's dilemma. In: PReMI (2005), S. K. Pal, S. Bandyopadhyay, and S. Biswas, (Eds.), vol. 3776 of Lecture Notes in Computer Science, Springer, 1–10

[24] Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **20** (December 1998), 359–392

[25] Kittler, J., Hatef, M., Duin, R. P. W.: Combining classifiers. In: Proceedings of the Sixth International Conference on Pattern Recognition (Silver Spring, MD, 1996), IEEE Computer Society Press, 897–901

[26] MacQueen, J. B.: Some methods for classification and analysis of multivariate observations. In: Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability (1967), L. M. L. Cam and J. Neyman, (Eds.), vol. 1, University of California Press, 281–297

[27] Mimaroglu, S., Erdil, E.: Obtaining better quality final clustering by merging a collection of clusterings. Bioinformatics **26 (20)** (2010), 2645–2646

[28] Nguyen, N., Caruana, R.: Consensus clusterings. In: ICDM (2007), IEEE Computer Society, 607–612

[29] Rezaee, M. R., Lelieveldt, B., Reiber, J.: A new cluster validity index for the fuzzy c-mean. Pattern Recognition Letters **19 (3–4)** (1998), 237–246

[30] Salvador, S., Chan, P.: Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In: ICTAI (2004), IEEE Computer Society, 576–584

[31] Schwarz, G.: Estimating the dimension of a model. Annals of Statistics **6** (1978), 461–464

[32] Strehl, A., Ghosh, J.: Cluster ensembles – a knowledge reuse framework for combining multiple partitions. Journal on Machine Learning Research (JMLR) 3 (December 2002), 583–617

[33] Topchy, A. P., Jain, A. K., Punch, W. F.: Clustering ensembles: models of consensus and weak partitions. IEEE Trans. Pattern Anal. Mach. Intell. **27 (12)** (2005), 1866–1881

[34] Tumer, K., Agogino, A. K.: Ensemble clustering with voting active clusters. Pattern Recognition Letters **29 (14)** (2008), 1947–1953

[35] Ultsch, A., Mörchen, F.: ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM. Technical Report No. 46, Dept. of Mathematics and Computer Science, University of Marburg, Germany (2005)

[36] Wolpert, D. H., Macready, W. G.: No free lunch theorem for optimization. IEEE Transactions on Evolutionary Computation **1 (1)** (1997)

[37] Yu, Z., Wong, H.-S., Wang, H.-Q.: Graph-based consensus clustering for class discovery from gene expression data. Bioinformatics **23 (21)** (2007), 2888–2896