Evidence accumulation in multiobjective data clustering

Julia Handl[1] and Joshua Knowles[2]

[1] Manchester Business School, University of Manchester, UK
julia.handl@mbs.ac.uk
[2] School of Computer Science, University of Manchester, UK
j.knowles@manchester.ac.uk

Abstract. Multiobjective approaches to data clustering return sets of solutions that correspond to trade-offs between different clustering objectives. Here, an established ensemble technique (evidence-accumulation) is applied to the identification of shared features within the set of clustering solutions returned by the multiobjective clustering method MOCK. We show that this approach can be employed to achieve a four-fold reduction in the number of candidate solutions, whilst maintaining the accuracy of MOCK's best clustering solutions. We also find that the resulting knowledge provides a novel design basis for the visual exploration and comparison of different clustering solutions. There are clear parallels with recent work on 'innovization', where it was suggested that the design-space analysis of the solution sets returned by multiobjective optimization may provide deep insight into the core design principles of good solutions.

1 Introduction

Data clustering is the problem of identifying groups (clusters) of similar data items within collections of unlabelled data. One of the key challenges in this respect is the mathematical description of a good cluster, which may then be used to define an actual clustering objective. Existing objective functions for data clustering typically make fairly strong assumptions about the properties of a good cluster, and therefore lack robustness towards data that are violating those assumptions. Recently, multiobjective approaches to data clustering have been introduced with the aim of optimizing not one but several clustering objectives simultaneously. It has been argued that this use of several objectives facilitates a more natural (and robust) definition of the clustering problem, and recent work has shown that the sets of optimal trade-off solutions generated by multiobjective clustering techniques do indeed contain solutions that improve upon the quality of the solutions obtained by optimizing a single clustering objective only [6, 11].

As the objectives used in multiobjective clustering are typically conflicting, even a single run of a multiobjective clustering method (for a given number of clusters) will return a set of different trade-off solutions. Some multiobjective clustering methods, such as Multiobjective Clustering with Automatic k-Determination (MOCK, [10, 11]) additionally generate solutions across a range of different numbers of clusters and, therefore, return solution sets that cover both a range of different numbers of clusters and different trade-offs between the clustering objectives. In practical applications, a user of a multiobjective clustering technique will select one (or a few) preferred solution from the final set of optimal trade-off solutions. Evidently, there is a strong need to support the user during this process of *model selection*, and dedicated approaches to this end have been proposed in the literature. For the multiobjective clustering method MOCK, we previously devised an automated technique of model selection [10] that selects a single most promising solution from the set of trade-off solutions. The technique is based on an analysis of the location of solutions in objective space relative to a background of unstructured 'control data' [see 17]. When applied to the multiobjective clustering technique MOCK, this approach has been shown to outperform more traditional techniques of model selection such as the Silhouette Width [15].

A potential criticism of MOCK's standard model selection approach [10] is the following: The analysis is based entirely in objective space and does not fully utilize the information captured by the approximation set as a whole. Recent research in the field of evolutionary multiobjective optimization has shown the potential value of identifying features in design space that are overrepresented within the approximation set returned by an EMO algorithm [1, 2]. This raises the question of whether further improvements in the accuracy, presentation and selection of multiobjective clustering solutions may be feasible by integrating the information provided from the entire set of optimal trade-off solutions.

Although not applied to the multiobjective clustering algorithm MOCK before, the idea of integrating sets of solutions is not novel to the field of clustering and has been adressed in the form of cluster ensemble techniques [7, 8, 16]. Cluster ensemble techniques typically operate on sets of cluster assignments that are returned by a range of clustering methods and attempt to integrate these labels into a single 'consensus clustering'. In this context, the technique of evidence accumulation has been shown to be particularly effective [7], and this is the method we will adopt in our work. Specifically, we aim to investigate whether evidence accumulation provides a suitable means of integrating the set of trade-off solutions returned by multiobjective data clustering, whether this leads to an improvement in solution accuracy, and whether this enables us to obtain a better understanding of the relationships between solutions and the features shared by different optimal trade-off solutions.

In the following (Section 2), we briefly review a number of key concepts related to this work. Section 3 describes the experimental setup, including details of the algorithms and the data sets employed. Section 4 reports our results and discusses the key findings from our experiments. Finally, Section 5 concludes.

2 Background

In this section, we first discuss the principles of model selection for data clustering. We then provide some background on multiobjective clustering, and consider the use of model selection in multiobjective clustering. Finally, ensemble techniques for data-clustering are reviewed.

2.1 Model selection

Model selection, i.e., the identification of the most suitable solution or algorithm parameter, is a fundamental problem in data clustering. When a single, deterministic clustering technique is used (and all available partitionings are obtained for the same set of input features), the problem reduces to that of identifying the number of clusters k in a data set. More generally, however, the problem of model selection will also include choices between different possible partitionings with the same number of clusters, such as different solutions returned (for the same k) by a non-deterministic method such as k-means, or the results returned (for the same k) by different algorithms.

Model selection in clustering has been addressed using a variety of different techniques [see 9, 12, for reviews]. One of the most common approaches to model selection is the evaluation of all clustering solutions using a specialized internal validation index and the subsequent selection of the top scoring solutions. These indices of cluster validation typically assess the balance between some measure of intra-cluster and inter-cluster variation, and prominent examples include the Silhouette Width [15], the Dunn index [3] and the DB-Index [9]. Alternative approaches to model selection consider the stability of the partitionings under re-sampling [14] or the relative quality of a partitioning compared to a partitioning obtained on unstructured data [17].

2.2 Multiobjective clustering with automatic k-determination

The multiobjective clustering method MOCK [10] is based on the evolutionary multiobjective algorithm PESA-II [5] and has been designed for the optimization of two different clustering criteria. The first of these, *overall deviation*, measures the compactness of clusters, whereas the second objective, *connectivity*, considers whether adjacent data items are placed in the same clusters. See [10] for formal definitions.

A single run of the multiobjective clustering method MOCK returns a set of solutions that correspond to different trade-offs between these two objectives. One of MOCK's parameters is an upper limit on the required number of clusters (typically, $k = 25$ is used), but apart from this, the number of clusters is kept open. Many of the solutions returned by MOCK therefore correspond to different numbers of clusters, in addition to providing different trade-offs between the clustering objectives.

2.3 Model selection in multiobjective clustering

As multiobjective approaches to data-clustering typically return a set of possible clustering solutions, some previous work on these methods considered automatic ways of selecting a single preferred clustering solution.

In MOCK, an integrated method of model selection is used [11] which works, briefly, as follows: Given a data set of interest, MOCK is first run to determine an initial set of optimal trade-off solutions. MOCK then produces several sets of 'control data', which are unstructured data sets that are generated randomly within the bounds of the original data set. MOCK determines a set of optimal trade-off solutions for each of these sets of control data. After a normalization of the objective values, the distances between the initial solutions and the solutions on the control data can be compared in objective space. The initial solution that is furthest away from the control points is selected as the best solution. The approach is described in more detail in [11].

In the context of multiobjective fuzzy clustering, a different approach to model selection has been described by Maulik et al. [13]. For the multiobjective data clustering method MOGA (which returns possible partitionings for a single, fixed number of clusters), the authors (ibid.) developed an approach that utilizes an analysis in decision space: they use a re-labelling strategy to maximize the overlap between all of MOGA's output partitionings, and to identify those data points that are consistently assigned to the same cluster (and also have a significant degree of membership with that cluster). The cluster labels of those points are then used as the class labels in the training of a support vector machine, which is applied to the prediction of cluster membership for all remaining data points. Using this approach, the method was shown to achieve an improvement in terms of the Silhouette Width of the final clustering solution, though no external validation of the clustering results was performed.

2.4 Ensemble techniques

Methods designed for the combination of the output of different clustering techniques are often referred to as ensemble methods. Similar to bagging and boosting in supervised classification [4], clustering ensembles are designed to improve the performance of clustering techniques by combining the results from several different runs, parameterizations or types of algorithms. Ensemble techniques typically operate on sets of cluster assignments (the outputs from clustering algorithms) only and do not consider the original input data. One of the best-known groups of ensemble techniques are the methods introduced by Strehl and Ghosh [16], which use the idea of hypergraphs to collect information from various partitionings; they then apply graph partitioning methods to obtain a final consensus clustering.

A relatively recent development in ensemble clustering is the technique of *evidence accumulation*, introduced by Fred and Jain [7]. Similarly, to Strehl and Ghosh's approaches [16], the method starts with the cluster assignments returned by all algorithms, but the algorithm then proceeds to count co-associations between all data items. This information is used to construct a new dissimilarity matrix, which can then be partitioned using a standard hierarchical clustering approach. The dendrogram returned by the hierarchical algorithm can be cut to obtain a pre-specified number of clusters. The resulting partitioning provides a new consensus clustering, and this approach has been shown to outperform ensembles based on graph partitioning.

For our purpose, which is the aggregation of the solutions returned by multiobjective clustering, the method of evidence accumulation is appealing, as (i) it appears to be one of the best ensemble techniques currently available; (ii) it can be used to combine partitionings with different numbers of clusters; and (iii) it provides an output with a straightforward and intuitive interpretation: the height of a branch directly reflects information about the minimum strength of co-association between data items within that branch.

3 Method

We experimentally explore the use of evidence accumulation for the aggregation of solutions in multiobjective clustering. First, we assess the quality of the final solutions returned from evidence accumulation on MOCK's solution sets, and compare the quality of these solutions to those obtained using alternative approaches. We then discuss the potential of evidence accumulation to help in the visualization of clustering solutions and to reduce the problem of model selection in multiobjective clustering.

3.1 Sets of clustering solutions

In addition to the solution sets returned by MOCK, we generate alternative sets of solutions using a range of established clustering techniques. This is done in order to compare the performance of evidence accumulation for inputs derived from a range of different methods. Overall, five different sets of clustering solutions are used:

– MOCK (**M**): This set contains the solutions returned by MOCK for $k \in [1, 25]$. For the data sets considered, the output set of MOCK typically contains between 80 to 120 solutions (also see Figure 3 in the Results section). MOCK is run using standard parameter settings as described in [11].
– k-means (**K**): This set contains the solutions returned from the standard *R* implementation for k-means for $k \in [1, 25]$ (i.e., the set contains 25 solutions in total).
– Average-link (**A**): This set contains the solutions returned from the standard *R* implementations of average-link hierarchical clustering for $k \in [1, 25]$ (i.e., the set contains 25 solutions in total).
– Single-link (**S**): This set contains the solutions returned from the standard *R* implementations of average-link hierarchical clustering for $k \in [1, 25]$ (i.e., the set contains 25 solutions in total).
– Combined (**C**): This set combines the solutions sets of k-means, average-link and single-link (above). Overall, this set therefore contains 75 solutions.

3.2 Evidence accumulation

The next step of the experiments is to process some of the above sets as follows: Each of the sets is, individually, used as the input to Fred and Jain [7]'s method of evidence accumulation. We then generate a new set of output solutions by applying the appropriate cuts to the dendrogram and generating partitionings for $k \in [1, 25]$.

As single-link and average-link are hierarchical (and deterministic) methods, the application of evidence accumulation to their output alone does not lead to any new clustering solutions. Sets of inputs based on their individual outputs only are therefore not used in these experiments. Evidence accumulation therefore generates three new sets of solutions only, which are denominated as MOCK with Evidence Accumulation (**MEvAcc**), k-means with Evidence Accumulation (**KEvAcc**) and Combined with Evidence Accumulation (**CEvAcc**), and contain 25 solutions each.

Evidence accumulation is implemented as described by Fred and Jain [7]. Given a set of input clustering solutions for a data set containing N items (e.g. from a single run of MOCK), the $N \times N$ co-association matrix is constructed as

$$C(i, j) = \frac{m_{ij}}{M},$$

where M gives the number of clustering solutions contained in the set, and m_{ij} indicates the number of times (within those M partitions) that data items i and j have been assigned to the same cluster. A new dissimilarity matrix is then obtained as $D(i, j) = 1 - C(i, j)$, and two different hierarchical clustering methods (single-link and average-link agglomerative clustering) are used to construct the consensus partitions of the data. In line with Fred and Jain [7], the results for single-link agglomerative clustering are consistently worse than the results for average-link agglomerative clustering, so results for this are not shown in the experimental section.

3.3 Solution selection methods

Using the sets of solutions generated in the previous stages, we further investigate whether evidence accumulation may present a suitable approach for model selection in multiobjective clustering. For this purpose, we compare a number of alternative techniques of model selection. The first of these is MOCK's established approach [11], which identifies a single partitioning based on distances (in objective space) to random control data.

As a second option, we explore the use of the solution sets returned by evidence accumulation: The output from evidence accumulation is, initially, a set of solutions that contains a single solution for each possible number of clusters (here, $k \in [1, 25]$). As a result, the spacing of solutions along the Pareto front is more even than the spacing in the fronts returned directly from multiobjective clustering (which usually contain several solutions for each value of k). Knee detection based on the local shape of the Pareto front may therefore become more feasible, and we test this by calculating the angles between triplets of adjacent clustering solutions, and selecting the 'middle' solution with the smallest angles as the final solution.

Finally, as a third option, we consider the fact that evidence accumulation uses a hierarchical clustering algorithm to partition the co-association matrix, and that its output is, therefore, best represented using a dendrogram. In previous work, Fred and Jain [7] suggest that branch length within this dendrogram can be used for model selection: they propose to identify the cut that eliminates the longest branch in the dendrogram and select the associated partitioning as the best solution. We explore the potential of this approach for the dendrograms returned from evidence accumulation on MOCK's clustering solutions.

3.4 Data sets

The techniques discussed above are compared using a test suite of data sets that contain multiple Gaussian clusters in various dimensions. These data sets are generated using the cluster generator described in [11] and available online. The parameterization of the generator is shown in Table 1. Data sets are generated in three and ten dimensions and contain four, six or eight Gaussian clusters. Ten different instances are generated for each combination of dimension and cluster number, resulting in a total of 60 different instances. Individual instances are denoted as Dd-Cc-no$I$, where D indicates the dimensionality of the data, C indicates the number of clusters and I is the index of the instance. All experimental results reported are obtained over 21 independent runs per algorithm per instance, and the Euclidean distance function is used in all experiments.

Table 1. Parameters of the synthetic data generator, where N_k gives the number of points in the kth cluster, μ_{kd} defines the mean of the kth cluster in the dth dimension and σ_{kd} defines the variance of the kth Gaussian cluster in the dth dimension. The parameters of individual clusters are generated randomly within the bounds shown below.

Min N_k	Max N_k	Max μ_{kd}	Min μ_{kd}	Min σ_{kd}	Max σ_{kd}
10	100	10	-10	0	$20\sqrt{D}$

3.5 Comparison metrics

A range of techniques are used to evaluate the quality of the solution sets and individual clustering solutions. First, a visualization of the sets of clustering solutions in bi-objective space is used to understand the actual effect of evidence accumulation. As we are dealing with sets of clustering solutions in bi-objective space, some of these results are summarized in the form of attainment fronts. Results are obtained over 21 runs for each data set, so the first and eleventh attainment front are employed to indicate top and median performance.

Furthermore, the agreement of the partitionings with the known cluster memberships is determined using an an external validation technique. The Adjusted Rand Index is used for this purpose, as it provides an established way of comparing partitionings with different numbers of clusters [12]. It returns values within the range $[0, 1]$, where a value of 1 indicates a perfect agreement with the known cluster memberships. During the evaluation of results, the Adjusted Rand Index is utilized in two different ways. For the comparison of solution sets, we are interested in evaluating the algorithms performance at generating high-quality solutions. Hence, the comparison focuses on the best clustering solution found within each solution set (i.e., the solution that scores highest with respect to the Adjusted Rand Index is identified directly). When comparing techniques for model selection, evaluation is based on the Adjusted Rand Index of the final (single) solution selected.

Finally, we also consider the sizes of the solution sets returned by the different techniques.

4 Results

Figure 1 shows the evaluation of the solution sets for a three-dimensional data set with eight clusters. This visualization in bi-objective space (using MOCK's clustering objectives) reveals an interesting phenomenon regarding the effect of evidence accumulation: For the solution sets generated by k-means or the combination of algorithms, evidence accumulation generates results that dominate the original solutions with respect to MOCK's clustering objectives. Unlike the original input solutions, the solutions resulting from evidence accumulation tend to be mutually non-dominated. This

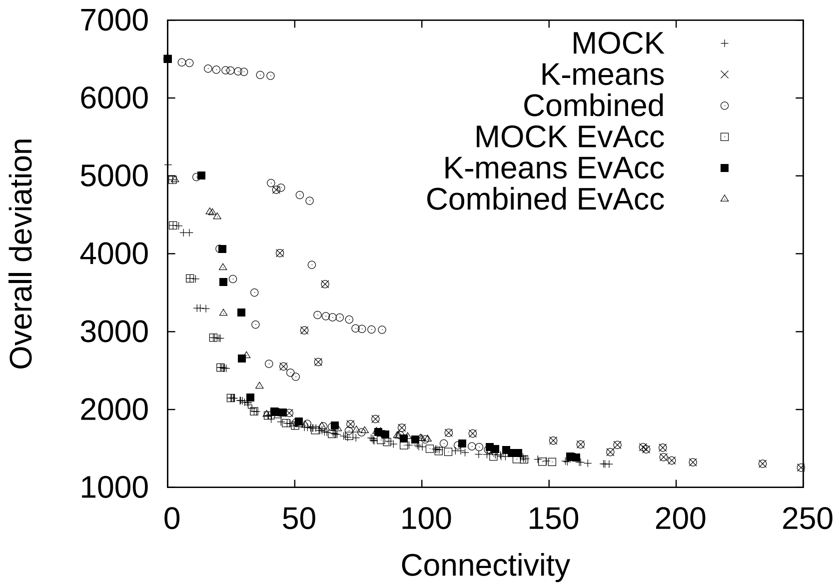

Fig. 1. Results for instance 3d-8c-no0. Sets of clustering solutions obtained by a single run of MOCK, MOCK with evidence accumulation (MOCK EvAcc), k-means, k-means with evidence accumulation (k-means EvAcc), the ensemble of three traditional algorithms (Combined), and the ensemble of three traditional algorithms with evidence accumulation (Combined EvAcc).

is surprising, as the objective of connectivity is not directly optimized by any of these algorithms. The results suggest that the technique of evidence accumulation produces solutions that implicitly optimize this measure. Interestingly, the same effect is not seen when evidence accumulation is applied to MOCK's solutions: Evidence accumulation does not generally produce solutions that dominate those contained in MOCK's original approximation front. This may be because MOCK's solutions are already close to optimal with respect to both objectives.

To provide a better idea of the stochastic variation in these results, Figure 2 shows the first and eleventh attainment fronts for all six algorithms on the same data set. It can be seen that there is no substantial difference in terms of the attainment of MOCK's solutions before and after evidence accumulation. On the other hand, it is clear that both sets of results dominate the solution sets returned by alternative techniques.

Next, we consider the size of the solution sets and the quality of the best solutions in terms of the known cluster memberships. Summary results over all 60 instances are show in Figures 3 and 4, in the form of boxplots. Consistent with the observations in objective space and the results in [7], the application of evidence accumulation results in improved solutions (compared to the original input solutions) for the use with k-means solutions. For the output of MOCK and the ensemble of algorithms, we see no such effect in terms of the accuracy of the best clustering solutions. For MOCK, this result is consistent with our observations in objective space: It seems that evidence accumulation is not able to improve upon the solutions returned by multiobjective clustering, which may be due to the strong performance of MOCK on these data sets.

We next investigate the size of the solution sets in Figures 3 and 4. From these data, it is evident that the application of evidence accumulation results in a significant (about four-fold) reduction in the size of MOCK's solution sets, which is an important advantage. The results also show that this reduction comes at no significant expense in terms of solution quality: in terms of the Adjusted Rand Index, the best solutions returned by both MOCK and MOCK EvAcc are usually comparable and reliably outperform the best solutions returned by the six alternatives considered.

We are further interested whether evidence accumulation will allow for more effective means of model selection, and Figure 5 shows the related comparisons. The performance of the three model selection techniques is mixed. While, overall, MOCK's original strategy shows the most consistent performance, the angle and the dendrogram-based technique show very good performances for some of the data sets. The angle and dendrogram-based techniques are conceptually different and exploit different types of information, which leads us to hope that, in future work, higher robustness may be achieved through the integration of both approaches. Compared to MOCK's established selection strategy, an important advantage of both of these approaches is reduced computational expense, as they do not rely on the costly generation and clustering of control data.

Finally, we consider how the information derived from evidence accumulation may be used to support a user in the exploration of the solution sets returned by a multiobjective clustering algorithm. Evidence accumulation captures valuable information about the frequency of co-assignment of different items, which is displayed in the resulting dendrogram. We suggest to use this dendrogram for the visualization of individual clustering solutions. In Figure 6, this concept is illustrated for MOCK's output on a four-cluster data set.

5 Conclusion

Evidence accumulation is a state-of-the-art ensemble technique that has been shown to provide an effective way of combining and improving the results of traditional clustering techniques. This manuscript investigates evidence accumulation as a means to support the post-processing of the clustering solutions returned by the multiobjective clustering method MOCK. On the data sets considered, we find that evidence accumulation does not improve the accuracy of MOCK's clustering solutions, but that it achieves a substantial reduction in the number of trade-off solutions to be considered (with no loss of accuracy). We further demonstrate how the knowledge generated by evidence accumulation may be used in the selection, visualization and analysis of the solutions returned by multiobjective clustering. Future work may look into the integration of evidence accumulation into MOCK's search, as well as the development of more robust approaches to solution selection.

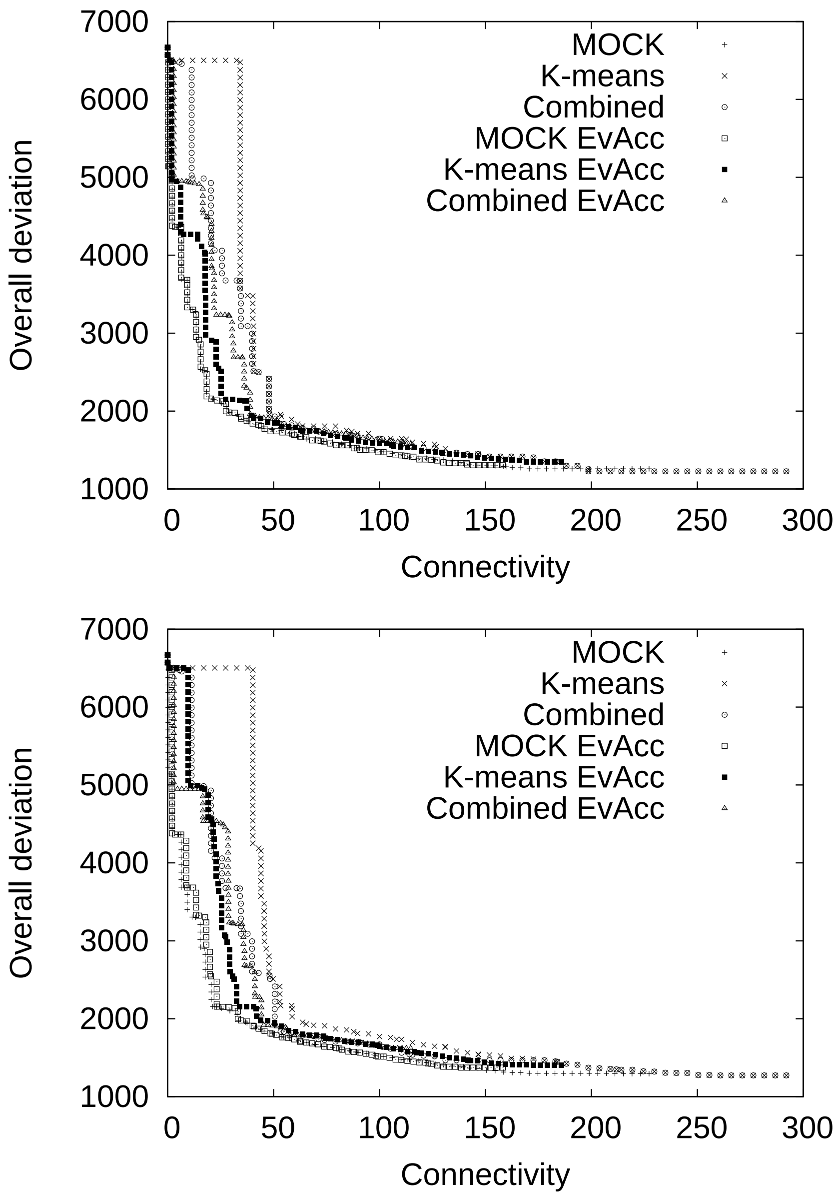

Fig. 2. Attainment fronts on instance 3d-8c-no0 for MOCK, MOCK with evidence accumulation (MOCK EvAcc), k-means, k-means with evidence accumulation (k-means EvAcc), the ensemble of three traditional algorithms (Combined), and the ensemble of three traditional algorithms with evidence accumulation (Combined EvAcc). (Top) First (best) attainment front. (Bottom) Eleventh (median) attainment front.

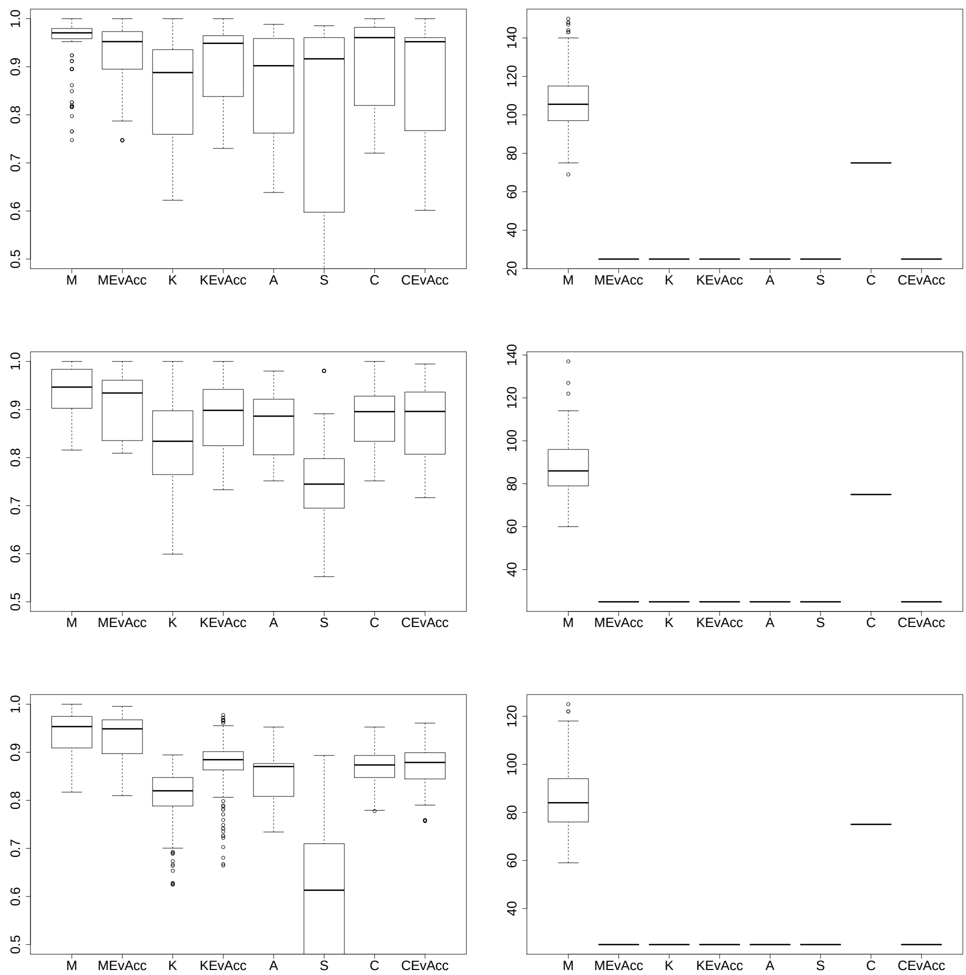

Fig. 3. Results for 21 runs each across ten different instances with three dimensions and (top) four clusters; (centre) six clusters; and (bottom) eight clusters. (Left) Adjusted Rand Index of the best solution in the final set of clustering solutions for each algorithm; (right) Number of solutions in the final set of clustering solutions. for each algorithm.

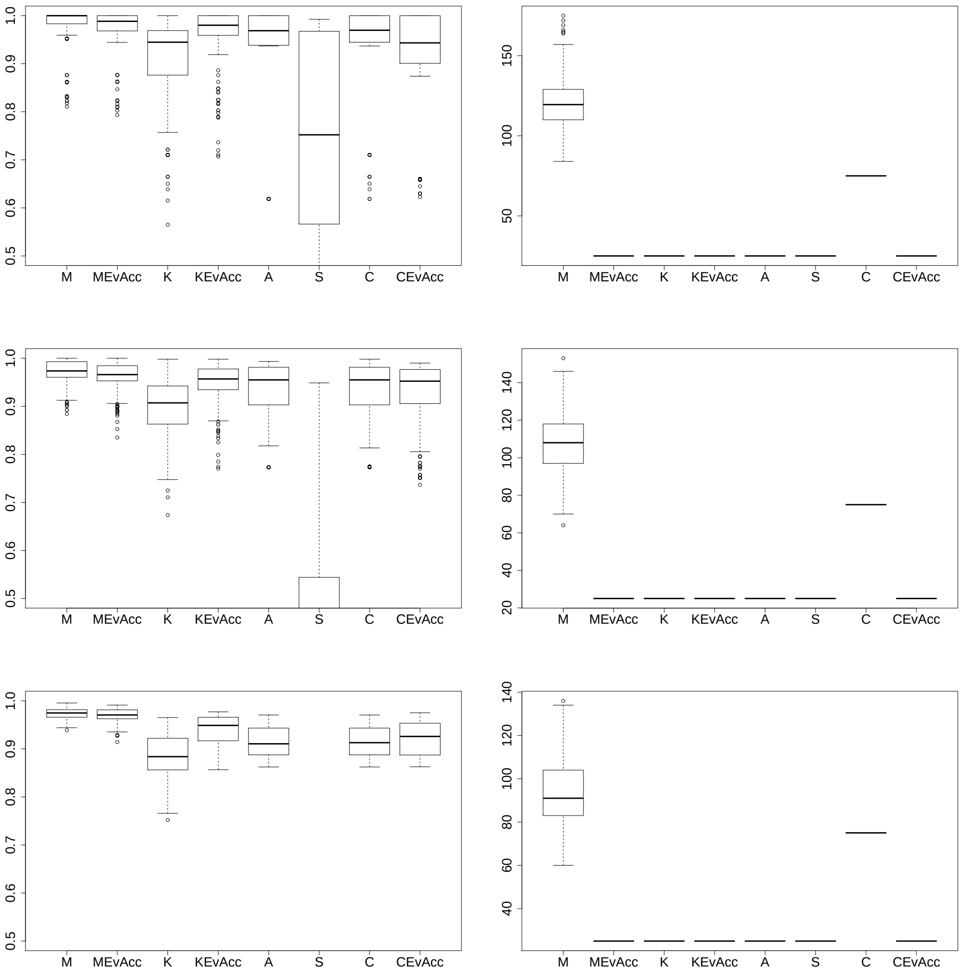

Fig. 4. Results for 21 runs each across ten different instances with ten dimensions and (top) four clusters; (centre) six clusters; and (bottom) eight clusters. (Left) Adjusted Rand Index of the best solution in the final set of clustering solutions for each algorithm; (right) Number of solutions in the final set of clustering solutions. for each algorithm.

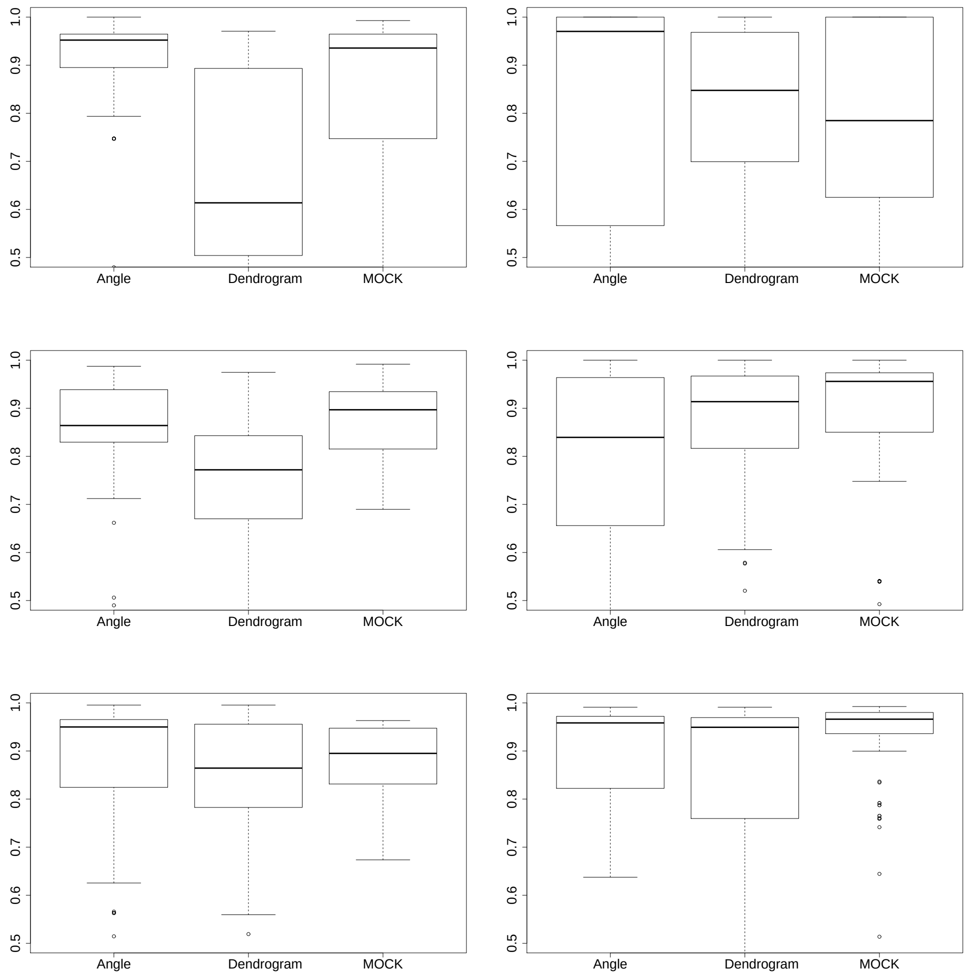

Fig. 5. Results for 21 runs each across ten different instances with (left) three dimensions; (right) ten dimensions. (Top) four clusters; (centre) six clusters; (bottom) eight clusters. Adjusted Rand Index of the solution selected by different methods of solution selection.

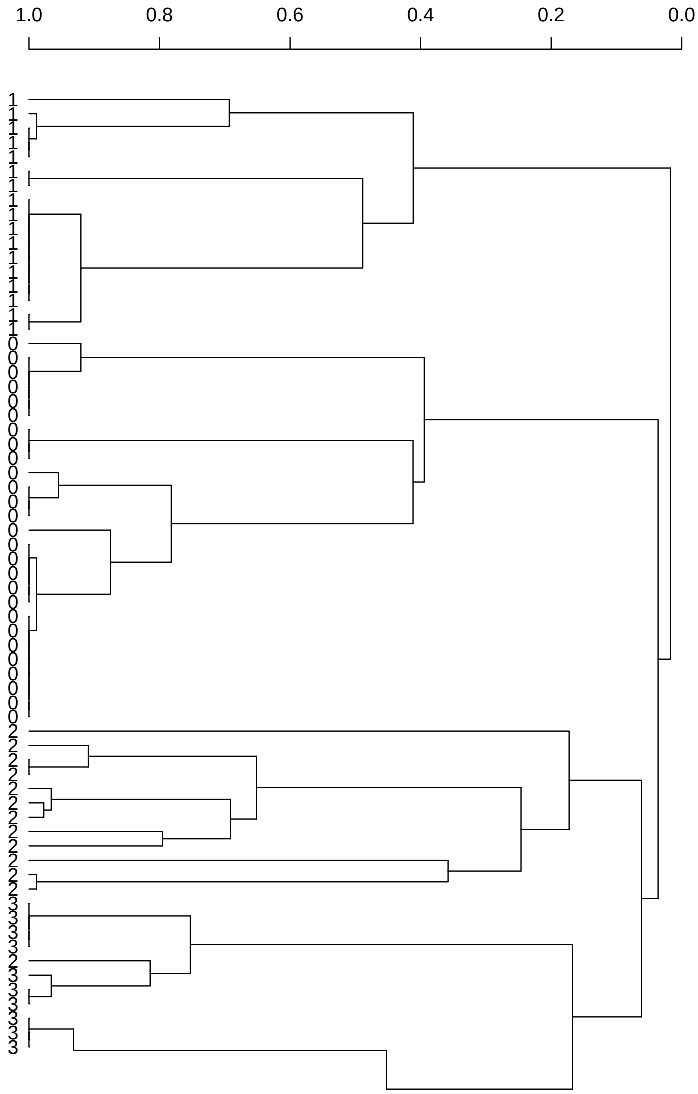

Fig. 6. Visualization of one of MOCK's clustering solutions for a sub-sampled instance 3d-4c-no0. The dendrogram structure is obtained based on evidence accumulation of all of MOCK's trade-off solutions, wheras the numbers displayed at the leaf nodes reflect the assignments made by a single, selected clustering solution. Using this visualization, a user can easily identify discrepancies between this particular solution and the 'majority opinion': here, the dendrogram is almost entirely consistent with the labelling provided by the selected solution (it can be seen that a cut of the dendrogram for $k = 4$ would result in an almost identical clustering solution), indicating that the particular solution is in strong agreement with the majority of solutions in MOCK's complete set of trade-off solutions. There is one discrepancy in the fourth cluster (note the single label of "2" within a series of "3"s), which highlights a data point that has been misclassified by the clustering solution selected. The visualization also helps in identifying data items for which there is particularly low or high uncertainty in the cluster assignment, e.g. the length of the branches in the dendrogram indicates that, overall, there is higher consensus in the assignments to clusters 0 and 1, relative to assignments to clusters 2 and 3. This provides additional information about the level of definition of individual cluster structures in the underlying data.

References

[1] S. Bandaru and K. Deb. Automated discovery of vital knowledge from Pareto-optimal solutions: First results from engineering design. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[2] S. Bandaru and K. Deb. Automated innovization for simultaneous discovery of multiple rules in bi-objective problems. In *Evolutionary Multi-Criterion Optimization*, pages 1–15. Springer, 2011.

[3] J.C. Bezdek and N.R. Pal. Cluster validation with generalized Dunn's indices. In *Artificial Neural Networks and Expert Systems, 1995. Proceedings., Second New Zealand International Two-Stream Conference on*, pages 190–193. IEEE, 1995.

[4] G. Brown, J. Wyatt, R. Harris, and X. Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20, 2005.

[5] D.W. Corne, N.R. Jerram, J.D. Knowles, and M.J. Oates. PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 283–290, 2001.

[6] M. Delattre and P. Hansen. Bicriterion cluster analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(4):277–291, 1980.

[7] A.L.N. Fred and A.K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions*, 27 (6):835–850, 2005.

[8] R. Ghaemi, M.N. Sulaiman, H. Ibrahim, and N. Mustapha. A survey: clustering ensembles techniques. In *Proceedings of Computer, Electrical, and Systems Science, and Engineering (CESSE)*, volume 38, pages 644–653, 2009.

[9] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145, 2001.

[10] J. Handl and J. Knowles. Exploiting the trade-off — the benefits of multiple objectives in data clustering. In *Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 3410 of *LNCS*, pages 547–560. Springer-Verlag, 2005.

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

[12] J. Handl, J. Knowles, and D. B. Kell. Computational cluster validation for post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, 2005.

[13] U. Maulik, A. Mukhopadhyay, and S. Bandyopadhyay. Combining Pareto-optimal clusters using supervised learning for identifying co-expressed genes. *BMC Bioinformatics*, 10(1): 27, 2009.

[14] V. Roth, T. Lange, M. Braun, and J.M. Buhmann. A resampling approach to cluster validation. In *COMPSTAT*, pages 123–128, 2002.

[15] P.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[16] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[17] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.