1

Multiobjective clustering and cluster validation

Julia Handl and Joshua Knowles

University of Manchester
jhandl@postgrad.manchester.ac.uk, jknowles@manchester.ac.uk

Summary. This chapter is concerned with unsupervised classification, that is, the analysis of data sets for which no (or very little) training data is available. The main goals in this data-driven type of analysis are the discovery of a data set's underlying structure, and the identification of groups (or clusters) of homogeneous data items — a process commonly referred to as cluster analysis.

Clustering relies on the use of certain criteria that attempt to capture those aspects that humans perceive as the properties of a good clustering solution. A variety of such criteria exist, many of which are partially complementary or even conflicting, and may favour different types of solutions. Like many other machine learning problems, clustering can therefore be considered as an intrinsically multiobjective optimization problem.

In this chapter, we consider two steps in the clustering process that may benefit from the use of multiple objectives. First, we consider the generation of clustering solutions and show that the use of two complementary clustering objectives results in an improved and more robust performance vis-a-vis single-objective clustering algorithms. Second, we consider the problem of model selection, that is, the choice of the best clustering solution out of a set of different alternatives, and investigate whether a multiobjective approach may also be beneficial vis-a-vis more traditional validation criteria.

1.1 Unsupervised classification

The increasing volume of data arising in the fields of document retrieval and bioinformatics are two prominent examples of a trend in a wide range of different research areas. Novel technologies (such as the Internet in document retrieval, microarray experiments in bioinformatics, physical simulations in scientific computing and many more) give rise to large 'warehouses' of data, which can only be handled and processed by means of computers. The efficient analysis and the generation of new knowledge from these masses of data requires the extensive use of data-driven inductive approaches. Data-driven techniques are also referred to as unsupervised techniques, and stand in contrast to supervised techniques, which require the presence of training data, that is, a (sufficiently large) set of data samples for which the correct classification is known. In the absence of training data, algorithms rely on

the presence of distinct structure in the data, that is, it must be hoped that a distance measure or a reduced feature space can be identified under which related data items cluster together in data space. This type of method is also referred to as clustering or unsupervised classification [7, 10, 16, 19].

1.1.1 Clustering

Informally, clustering is concerned with the division of data into homogeneous subgroups, subject to the following two aims: data items within one cluster should be similar to each other, while those within different clusters should be dissimilar. Formally, the *clustering problem* can be defined as an optimization problem[1] [3]:

Definition 1.1: The clustering problem

INSTANCE: A finite set X, a distance measure $\delta(i,j) \in \Re_0^+$ for $i,j \in X$, a positive integer K, and a criterion function $J(C, \delta(\cdot,\cdot))$ on a K-partition $C = \{C_1, \ldots, C_K\}$ of X and measure $\delta(\cdot,\cdot)$.

OPTIMIZATION: Find the partition of X into disjoint sets $C_1, \ldots, C_K$ that maximises the expression $J(C, \delta(\cdot,\cdot))$.

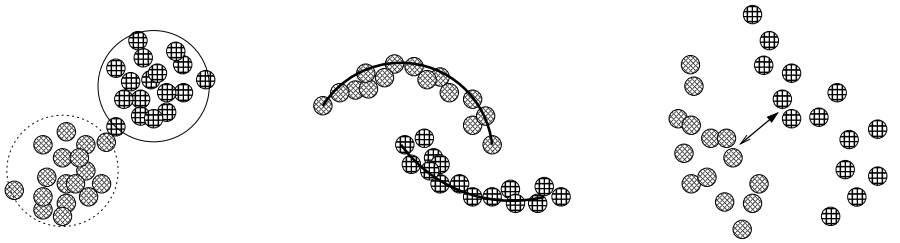
The number $R(K,N)$ of possible solutions for the division of a data set of size N into K partitions is given by the Stirling number of the second kind:

$$R(K,N) = \frac{1}{K!}\sum_{i=1}^{K}(-1)^{K-i}\binom{K}{i}(i)^N \approx \frac{K^N}{K!}$$

Hence, even with a fixed number of partitions K, the search space for the clustering problem grows exponentially and cannot be scanned exhaustively even for medium sized problems. Indeed, the clustering problem is known to be NP-hard in many of its forms [3].

Definitions of the clustering problem vary in the optimization criterion J and the distance function $\delta(\cdot,\cdot)$ used. Unfortunately, choosing an optimization criterion, that is, grasping the intuitive notion of a cluster by means of an explicit definition, is one of the fundamental dilemmas in clustering [9, 20]. While there are several valid properties that may be ascribed to a good clustering, these may be partially conflicting and/or inappropriate in a particular context (see Figure 1.1). Yet, most existing clustering methods attempt, explicitly or otherwise, to optimize just one such property — and through this choice they make a priori assumptions on the type of clusters that can later be identified. This confinement to a particular clustering criterion is one of the reasons for the fundamental discrepancies observable between

[1] Without loss of generality, we here assume maximization.



A: Compactness B: Connectedness C: Spatial separation

Fig. 1.1. There are several valid properties that may be ascribed to a good partitioning, but these are partly in conflict and are generally difficult to express in terms of objective functions. Despite this, existing clustering criteria/algorithms do fit broadly into three fundamental categories:

(1) Compactness. This concept is generally implemented by keeping intra-cluster variation small. This category includes algorithms like K-means [21], average link agglomerative clustering [32] or model-based clustering approaches [22]. The resulting methods tend to be very effective for spherical or well-separated clusters, but they may fail to detect more complicated cluster structures [7, 10, 16, 19].

(2) Connectedness. This is a more local concept of clustering based on the idea that neighbouring data items should share the same cluster. Algorithms implementing this principle are density-based methods [1, 8] and methods such as single link agglomerative clustering [32]. They are well-suited for the detection of arbitrarily shaped clusters, but can lack robustness when there is little spatial separation between clusters.

(3) Spatial separation. Spatial separation on its own is a criterion that gives little guidance during the clustering process and can easily lead to trivial solutions. It is therefore usually combined with other objectives, most notably measures of compactness or balance of cluster sizes. The resulting clustering objectives can be tackled by general-purpose meta-heuristics (such as simulated annealing, tabu search and evolutionary algorithms [2, 25]).

Examples of data sets exhibiting compactness, connectedness and spatial separation, respectively, are shown above. Clearly, connectedness and spatial separation are related (albeit opposite) concepts. In principle, the cluster structure in the data sets B and C can be identified by a clustering algorithm based on either connectedness or on spatial separation, but not by one based on compactness.

the solutions produced by different algorithms and will cause a clustering method to fail in a context where the criterion employed is inappropriate.

In practical data-mining scenarios, researchers attempt to circumvent this problem through the application and comparison of multiple clustering algorithms [18], or through the a posteriori combination of different clustering results by means of ensemble methods [28, 31].

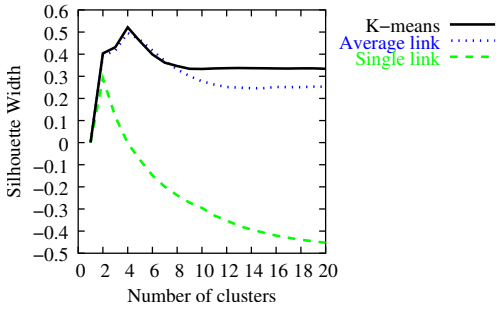


Fig. 1.2. If the structure of a data set and the best number of clusters is not known, a typical approach is to run several clustering algorithms for a range of numbers of clusters. Here, K-means, average link and single link agglomerative clustering have been run on a four-cluster data set (Square3) for $K \in [1, 20]$. The solutions are evaluated under a validation measure, here, the Silhouette Width [26], which takes both cluster compactness and cluster separation into account. For each algorithm, clear local maxima within the corresponding curve (here, at $K = 4$ for K-means and average link and at $K = 2$ for single link), are considered as good solutions. The algorithm resulting in the highest validation values (here, K-means) may be selected as the most suitable clustering method. This way of choosing one partitioning from a set of different partitionings is frequently referred to as model selection.

1.1.2 Model selection

Determining the most appropriate number of clusters K for a given data set is a second important challenge in clustering. Automated approaches to the determination of the number of clusters frequently work by selecting the most appropriate clustering from a range of partitionings with different Ks, a process often referred to as model selection (see Figure 1.2). Here, the partitioning that is most appropriate is again defined by means of a clustering criterion. Formally, the process of model selection considered in this chapter can be defined as follows:[2]

Definition 1.2: Model selection

INSTANCE: A finite set X, a distance measure $\delta(i, j) \in \Re_0^+$ for $i, j \in X$, a set of M partitionings $\{P_1, P_2, \ldots, P_M\}$, and a criterion function $J(P, \delta(\cdot, \cdot))$ on a partition P of X and measure $\delta(\cdot, \cdot)$.

OPTIMIZATION: Find the partition $C \in \{C_1, C_2, \ldots, C_M\}$, for which the expression $J(C, \delta(\cdot, \cdot))$ is maximal.

[2] Without loss of generality, we here assume maximization.

Again, fundamentally different clustering objectives can be used in this process, and the final result may vary strongly depending on the specific objective employed. There is a consensus that good partitionings tend to perform well under multiple complementary clustering criteria and the best clustering is therefore commonly determined using linear or non-linear combinations of different clustering criteria.

1.1.3 Scope of this work

Evidently, the choice of an appropriate clustering criterion is of major importance in both clustering and model selection. Unfortunately, it is also clear that a secure choice requires knowledge of the underlying data distribution, which is, in most cases, not known a priori.

In this article, we contend that the framework of Pareto optimization can provide a principled way to deal with this issue. Multiobjective Pareto optimization allows the simultaneous optimization of a set of complementary clustering objectives, which abolishes the need for multiple runs of different clustering algorithms, and is more general than the fixed linear or non-linear combination of individual objectives (see Figure 1.3). We therefore expect the direct optimization of multiple objectives to be beneficial both in the context of clustering and model selection.

In the following, we provide evidence to support this hypothesis in several steps. In Section 2, we describe a multiobjective clustering algorithm, MOCK, which simultaneously optimizes two different complementary clustering objectives. Experimental results show that the algorithm has a significantly improved classification performance compared to traditional single-objective clustering methods, and is highly robust over a range of different data properties. In Section 3, we consider model selection and discuss how the output of MOCK can be directly used to assess the quality of individual clustering results. Experimental results demonstrate the high performance of the resulting multiobjective validation approach, and discuss performance differences with respect to single-objective validation techniques. Finally, Section 4 concludes.

1.2 Multiobjective clustering

In order to develop a clustering algorithm that simultaneously considers several complementary aspects of clustering quality, we embrace the framework of Pareto optimization. Specifically, we employ a multiobjective evolutionary algorithm (MOEA) to optimize several clustering objectives, and to obtain a set of trade-off solutions, which represent a good approximation to the Pareto front. The resulting multiobjective clustering algorithm is named "MultiObjective Clustering with automatic K-determination" (MOCK, [13, 14, 15]).

1.2.1 PESA-II

MOCK is based on an existing MOEA, PESA-II. PESA-II [4, 5] is a well-known algorithm in the evolutionary multiobjective optimization literature, and has been

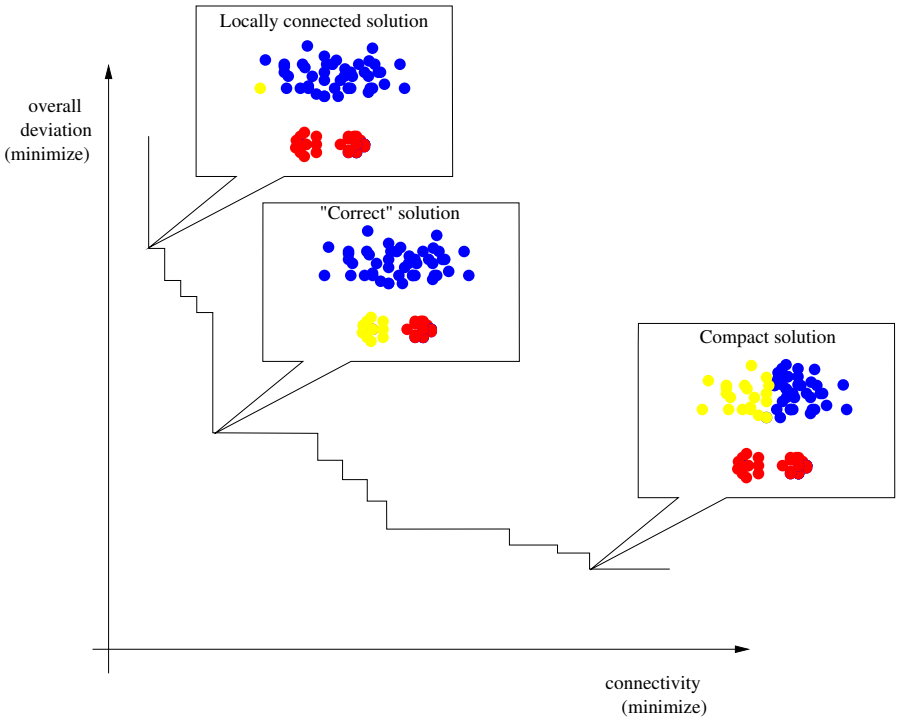


Fig. 1.3. Clustering can be considered as a multiobjective optimization problem. Instead of finding a single clustering solution, the aim in a multiobjective clustering scenario is to find an approximation to the Pareto front, that is the set of partitionings that are Pareto optimal with respect to the objectives optimized. Here, two objectives are to be minimized and the approximation set is represented by the corresponding attainment surface, which is the boundary in the objective space separating those points that are dominated by or equal to at least one of the data points in the approximation set from those that no data point in the approximation set dominates or equals. The figure highlights three solutions within this approximation set. The solutions to the top left tend to be locally connected but not compact, whereas those to the bottom right tend to be highly compact. The correct solution corresponds to a compromise between the two objectives, and can therefore only be found through the optimization of both objectives. For sake of clarity, the approximation set in this example only contains solutions for $K = 3$. More generally, the number of clusters can also be kept dynamic — in this case an approximation set is obtained in which the number of clusters varies along the Pareto front.

used in comparison studies by several researchers. In PESA-II, two populations of solutions are maintained: an internal population, IP of fixed size, and an external population EP, of non-fixed but limited size. The purpose of EP is to exploit good solutions: to this end it implements elitism by maintaining a large and diverse set of nondominated solutions. The internal population's job is to explore new solutions, and achieves this by the standard EA processes of reproduction and variation (that is, recombination and mutation). Selection occurs at the interface between the two populations, primarily in the update of EP.

The solutions in EP are stored in 'niches', implemented as a hypergrid in the objective space. A tally of the number of solutions that occupy each niche is kept and this is used to encourage solutions to cover the whole objective space, rather than bunch together in one region. To this end, nondominated solutions that try to enter a full EP can only do so if they occupy a less crowded niche than some other solution. Moreover, when the internal population of each generation is constructed from EP, they are selected uniformly from among the populated niches — thus highly populated niches do not contribute more solutions than less populated ones.

An important advantage of PESA-II is that this niching policy uses an adaptive range equalization and normalization of the objective function values. This means that difficult parameter tuning is avoided, and objective functions that have very different ranges can be readily used. PESA-II can also handle any number of objective functions. For further details on PESA-II, the reader is referred to [6].

1.2.2 Details of MOCK

The application of PESA-II to the clustering problem requires the choice of

- two or more objective functions,
- a suitable genetic encoding of a partitioning,
- one or more genetic variation operators (e.g. mutation and/or crossover),
- and an effective initialization scheme.

These choices are non-trivial and are crucial for the performance and particularly the scalability of the algorithm: many encodings work well for data sets with a few hundred data points, but their performance breaks down rapidly for larger data sets. The design of an effective EA for clustering requires a close harmonization of the encoding, the operators and the objective functions, which permits to narrow down the search space and guide the search effectively. Following extensive experiments using a range of different encodings, operators and objective functions, an effective combination of encoding, operators and objective functions was derived, the details of which are explained in the following.

Objective functions

For the clustering objectives, we are interested in selecting optimization criteria that reflect fundamentally different aspects of a good clustering solution. From the groups identified in Figure 1.1, we therefore select two types of complementary objectives:

one based on compactness, the other one based on connectedness of clusters. We refrain from using a third objective based on spatial separation, as the concept of spatial separation is intrinsic (opposite) to that of connectedness of clusters.

In order to express cluster compactness we compute the *overall deviation* of a partitioning. This is simply computed as the overall summed distances between data items and their corresponding cluster centre:

$$Dev(C) = \sum_{C_k \in C} \sum_{i \in C_k} \delta(i, \mu_k),$$

where C is the set of all clusters, μ_k is the centroid of cluster C_k and $\delta(.,.)$ is the chosen distance function. As an objective, overall deviation should be minimized. The criterion is strongly biased towards spherically shaped clusters.

As an objective reflecting cluster connectedness, we use a measure, connectivity, which evaluates the degree to which neighbouring data-points have been placed in the same cluster. It is computed as

$$Conn(C) = \sum_{i=1}^{N} \left(\sum_{j=1}^{L} x_{i,nn_i(j)} \right), \text{where } x_{r,s} = \begin{cases} \frac{1}{j} \text{ if } \nexists C_k : r \in C_k \wedge s \in C_k \\ 0 \text{ otherwise,} \end{cases}$$

$nn_i(j)$ is the jth nearest neighbour of datum i, and L is a parameter determining the number of neighbours that contribute to the connectivity measure. As an objective, connectivity should be minimized. This criterion captures local densities — it can therefore detect arbitrarily shaped clusters, but is not robust toward overlapping clusters.

Overall deviation (like our mutation operator, see below) requires the one-off computation of the nearest neighbour lists of all data items in the initialization phase. Subsequently, both objectives, overall deviation and connectedness, can be efficiently computed in linear time.

An important aspect in the choice of these objective functions is their potential to balance each other's tendency to increase or decrease the number of clusters. While the objective value associated with overall deviation necessarily improves with an increasing number of clusters, the opposite is the case for connectivity. The interaction of the two is important in order to explore sensible parts of the solution space, and not to converge to trivial solutions (which would be N singleton clusters for overall deviation and only one cluster for connectivity).

Encoding

For the encoding, we employ the locus-based adjacency representation proposed in [23]. In this graph-based representation, each individual g consists of N genes $g_1, \ldots, g_N$, where N is the size of the clustered data set, and each gene g_i can take allele values j in the range $\{1, \ldots, N\}$. Thus, a value of j assigned to the ith gene, is then interpreted as a link between data items i and j: in the resulting clustering

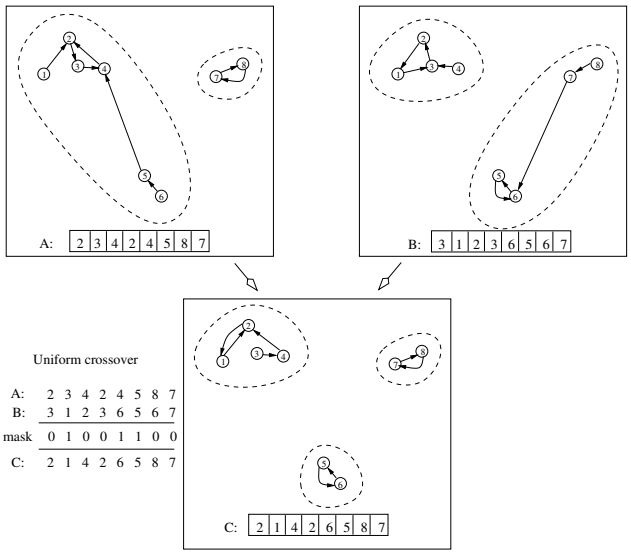


Fig. 1.4. Two parent partitionings, their graph structure, and their respective genotypes, A and B are shown. A standard uniform crossover of the genotypes yields the child C, which has inherited much of its structure from its parents, but differs from both of them.

solution they will be in the same cluster. The decoding of this representation requires the identification of all subgraphs. All data items belonging to the same subgraph are then assigned to one cluster. Note that, using a simple backtracking scheme, this decoding step can be done in linear time [13].

The locus-based adjacency encoding scheme has several major advantages for our application. Most importantly, there is no need to fix the number of clusters in advance, as it is automatically determined in the decoding step. Hence, we can evolve and compare solutions with different numbers of clusters in just one run of the GA. Furthermore, the representation is well-suited for use with standard crossover-operators such as uniform, one-point or two-point crossover. In more traditional encodings for clustering these straightforward crossover operators are usually highly disruptive and therefore detrimental for the clustering process. In a link-based encoding, in contrast, they effortlessly implement merging and splitting operations on individual clusters, while maintaining the remainder of the partitioning (see Figure 1.4).

Uniform crossover

We choose the uniform crossover [29] in favour of one- or two-point because it is unbiased with respect to the ordering of genes and can generate any combination of alleles from the two parents (in a single crossover event) [33].

Neighbourhood-biased mutation operator

While the encoding results in a very large search space with N^N possible combinations, a suitable mutation operator can be employed to significantly reduce the size of the search space. We use a restricted *nearest neighbour mutation* where each data item can only be linked to one of its L nearest neighbours. Hence, $g_i \in \{nn_{i1}, \ldots, nn_{iL}\}$, where nn_{il} denotes the lth nearest neighbour of data item i. This reduces the extent of the search space to just L^N. Note that the nearest neighbour list can be precomputed in the initialization phase of the algorithm.

The properties of the encoding can additionally be employed to bias the mutation probabilities of individual genes. Intuitively, 'longer links' in the encoding are expected to be less favourable, that is, a link $i \to j$ with $j = nn_{il}$ may be preferred over a link $i \to j^*$ with $j^* = nn_{ik}$ if $l < k$. This can be used to bias the mutation probability of individual links $i \to j$, which we now define as

$$p_m = \frac{1}{N} + \left(\frac{l}{N}\right)^2,$$

where $j = nn_{il}$ and N is the size of the data set.

Initialization

Our initialization routine is based on the observation that different clustering algorithms tend to perform better (find better approximations) in different regions of the Pareto front [14]. In particular, algorithms consistent with connectivity tend to generate close to optimal solutions in those regions of the Pareto front where connectivity is low, whereas algorithms based on compactness perform well in the regions where overall deviation has significantly decreased. We therefore use an initialization based on two different single-objective algorithms, in order to obtain a good initial spread of solutions and a close initial approximation to the Pareto front.

Solutions performing well under connectivity are generated using minimum spanning trees (MSTs). The use of MSTs has the advantage that the links present in a given MST can be directly translated as the encoding of individual solutions. Solutions performing well under overall deviation are generated using the K-means algorithm. The translation of these flat (that is, non-hierarchical) partitionings to the encoding of individual solutions is more involved, as described below.

Generation of interesting MST solutions

For a given data set, we first compute the complete MST using Prim's algorithm [34]. As the complete MST corresponds to a one-cluster solution, we are then interested in obtaining a range of good solutions with different numbers of clusters. Simply removing the largest links from this MST does not yield the desired results: in the

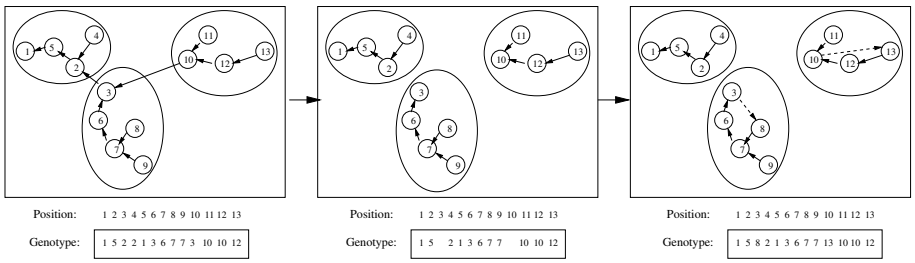


Fig. 1.5. Construction of an MST-similar solution from a given K-means solution. Starting from the original MST solution, all links that cross cluster boundaries (defined by the three-cluster K-means solution indicated here by the ellipses) are removed. The missing links are then replaced by a link to a randomly chosen neighbour with $i = nn_{il} \wedge l \leq L$.

absence of spatially separated clusters the method tends to isolate outliers so that many of the solutions generated are highly similar.

In order to avoid this effect, we employ a definition of interestingness that distinguishes between 'uninteresting' links whose removal leads to the separation of outliers, and 'interesting' links whose removal leads to the discovery of real cluster structures.

Definition 1.1: A link $i \rightarrow j$ is considered as *interesting*, iff $i = nn_{jl} \wedge j = nn_{ik} \wedge l > L \wedge k > L$, where L is a parameter. Its *degree of interestingness* is $d = \min(l, k)$.

Definition 1.2: A clustering solution C is considered as *interesting*, if it can be deduced from the full MST through the removal of interesting links only.

For a given data set, a set of interesting MST-derived solutions can then be constructed as follows. In a first step, all I interesting links from the MST are detected and are sorted by their degree of interestingness. Using this sorted list, a set of clustering solutions is then constructed: for $n \in [0, \min(I, 0.5 \times \text{fsize})]$, where fsize is the total number of initial solutions, clustering solution C_n is generated by removing the first n interesting links. The missing links are then replaced by a link to a randomly chosen neighbour j with $j = nn_{il} \wedge l \leq L$.

Generation of K-means solutions

Next, we consider the generation of K-means solutions. We start by by running the K-means algorithm (for 10 iterations) for different numbers of clusters $K \in [2, fsize - (\min(I, 0.5 \times fsize) + 1)]$. The resulting partitionings are then converted to MST-based genotypes as illustrated in Figure 1.5. The preservation of a high degree of MST information (within clusters) at this stage is crucial for the quick convergence of the algorithm. Note that the numbers of clusters obtained as the final phenotypes are not pre-defined and can increase or decrease depending on the structure of the underlying MST.

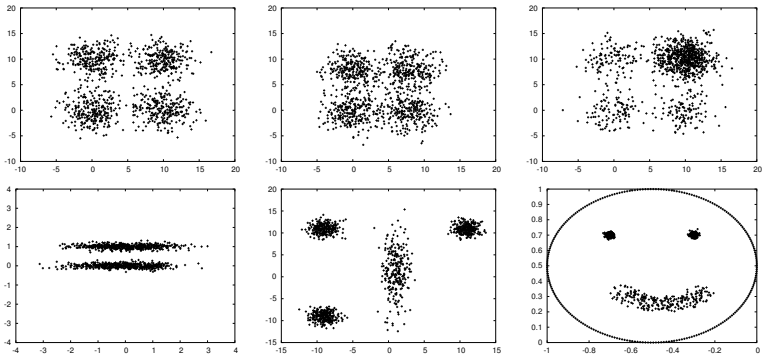


Fig. 1.6. Examples of data sets exhibiting different properties. The clusters in these data sets (with the exception of two clusters in Smile) are described by Normal Distributions, and, in every run, a new instance is sampled from these distributions (hence, all instances differ). Top row (from left to right): Square1, Square3 and Sizes3. Bottom row (from left to right): Long1, Triangle1 and Smile1.

1.2.3 Experimental setup

MOCK generates a set of different clustering solutions, which correspond to different trade-offs between the two objectives and to different numbers of clusters. While this provides a large set of solutions to chose from (typically more than one solution for each K), this experimental section primarily focuses on MOCK's 'peak' performance at generating high-quality clustering solutions. In order to assess this, we evaluate the quality of the best solution present in the final Pareto set. In order to put MOCK's results into perspective, we compare to three single-objective algorithms, which are run for a range of different numbers of clusters $K \in [1, 50]$. Again, only the best solution within this cluster range is selected (note that, on many data sets, the number of clusters of the best solution may be quite different from the known 'correct' number of clusters, e.g. because of the isolation of outliers).

We compare these algorithms on a range of data sets exhibiting different difficult data properties including cluster overlap (the Square series), unequally sized clusters (the Sizes series) and elongated clusters of arbitrary shape (the Long, Smile and Triangle series). See Figure 1.6 for an illustration of some of these data sets, and [12] for a detailed description.

Clustering quality is objectively evaluated using the Adjusted Rand Index, an external measure of clustering quality. External indices evaluate a partitioning by comparing to a 'gold standard', that is, the true class labels. Compared to other external indices, the Adjusted Rand Index has the strong advantage that it takes into account biases introduced due to the distribution of class sizes and differences in the number of clusters. This is of particular importance, as we compare clustering results across a range of different numbers of clusters.

Table 1.1. Number of clusters and quality of the best solution (in terms of the Adjusted Rand Index) generated by MOCK, K-means, Average link and Single link on a number of artificial data sets exhibiting different cluster properties (averages over 50 runs). The best and second best performer are highlighted in bold and italic face respectively.

Problem		MOCK		K-means		Average link		Single link	
Name	K	K	Rand	K	Rand	K	Rand	K	Rand
Square1	4	4.12	*0.96342*	4	**0.96505**	4.16	0.93701	37.76	0.82555
Square2	4	4.2	*0.92661*	4	**0.93574**	4.24	0.89713	40.12	0.45803
Square3	4	4.42	*0.86767*	4	**0.88436**	4.12	0.82932	41.32	0.10803
Sizes1	4	4.24	*0.96465*	4	**0.96663**	4.32	0.94117	36.22	0.84291
Sizes2	4	4.16	**0.97109**	4	*0.96532*	4.28	0.95103	35.64	0.82880
Sizes3	4	4.14	**0.97575**	4	*0.96336*	4.26	0.96008	34.94	0.87194
Long1	2	2	**0.99984**	5	0.35544	7.64	0.47172	3.48	*0.99570*
Long2	2	2	**0.99992**	4.88	0.33936	7.84	0.45719	3.42	*0.99595*
Long3	2	2.02	**0.99965**	4.42	0.21742	7.32	0.37855	3.32	*0.99612*
Smile1	4	4	**1**	30.48	0.74404	11.98	0.80083	4	**1**
Smile2	4	4	**1**	27.4	0.55456	10.54	0.75607	4	**1**
Smile3	4	4	**1**	33.48	0.34918	11.22	0.38256	4	**1**
Triangle1	4	4	**1**	4	0.95800	4.82	0.99259	4.14	*0.99979*
Triangle2	4	4	**1**	4	0.88607	5.02	*0.95435*	24.4	0.94976

Implementation details for the individual algorithms and the Adjusted Rand Index are given in the Appendix.

1.2.4 Experimental results

Experimental results confirm that MOCK is indeed robust towards the range of different data properties studied, and it clearly outperforms traditional single-objective clustering algorithms in this respect.

Table 1.1 shows the results of the comparison of MOCK to K-means, average link and single link. The results demonstrate that all single-objective algorithms perform well for certain subsets of the data, but that their performance breaks down dramatically for those data sets violating the assumptions made by the employed clustering criterion. As can be expected, K-means and agglomerative clustering perform very well for spherically shaped clusters but fail to detect clusters of elongated shapes. Single link detects well-separated clusters of arbitrary shape, but fails for 'noisy' data sets, such as the Sizes and Square series. Here, frequently, the failure of an algorithm becomes evident not only through a low Adjusted Rand Index, but also through a dramatic increase in the number of clusters of the best solution.

MOCK, in contrast, shows a consistently good performance across the entire range of data properties. It is the best performer on ten out of fourteen of the data sets. On the remaining four data sets it is the second best performer with only little

difference to the best performer K-means. The number of clusters associated with the best solution is usually very close to the real number of clusters in the data set.

For additional experimental results, including a comparison to an advanced ensemble technique, and results on complex high-dimensional and real test data with large numbers of clusters, the reader is referred to [12, 15]. All of these experiments confirm the high performance of the multiobjective clustering approach.

1.3 Multiobjective model selection

After dealing with the generation of clustering solutions in the previous section, we now turn our interest to model selection, that is the identification of one or several promising solutions from a large set of given clustering solutions. Model selection is particularly relevant for multiobjective clustering, as the algorithm does not return a single solution, but a set of solutions representing an approximation to the Pareto front. The individual partitionings in this approximation set correspond to different trade-offs between the two objectives but also consist of different numbers of clusters. While this may be a very useful feature under certain circumstances (e.g. human experts may find it preferable to have the opportunity to choose from a set of clustering solutions, to have the opportunity to analyze several alternative solutions and bring to bear any specialized domain expertise available), other applications may require the automatic selection of just one 'best' solution. In this section, we therefore introduce a method for identifying the most promising clustering solutions in the candidate set. We will also show how this methodology, originally developed for MOCK, can also be applied to analyze the output of other clustering algorithms.

1.3.1 Related work

The approach we have developed is inspired by Tibshirani et al's Gap statistic [30], a statistical method to determine the number of clusters in a data set. The Gap statistic is based on the expectation that the most suitable number of clusters shows in a significant 'knee' when plotting the performance of a clustering algorithm (in terms of a selected internal validation measure) as a function of the number of clusters. As internal validation measures are generally biased by the number of clusters (they show an increasing/decreasing trend that is solely due to a change in the number of clusters), the 'knee' can be best identified in a normalized plot, that is a performance plot that takes out the bias resulting purely from a change in the number of clusters. Tibshirani et al realize this by generating a number of reference partitionings for random data. From the normalized performance curve they then identify the smallest number of clusters for which the gain in performance is not higher than would be expected for random data.

1.3.2 Proposed approach

Several aspects of this idea can be carried over to our case of two objectives. Intuitively, we equally expect the structure of the data to be reflected in the shape of

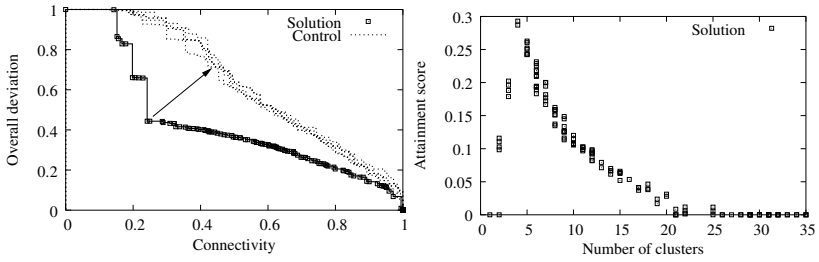


Fig. 1.7. (left) Solution and control reference fronts for a run of MOCK on the *Square1* data set. The solution with the largest minimum distance to the reference fronts is indicated by the arrow and corresponds to the desired $K = 4$ cluster solution. (right) Attainment scores for the *Square1* data set. Plot of the scores as a function of K. The global maximum at $K = 4$ is clearly visible.

the Pareto front. From the two objectives employed, overall deviation decreases with an increasing number of clusters, whereas connectivity decreases. Hence, generally, when considering two solutions with $K = k$ and $K = k + 1$ respectively (where $k \in [1, N]$, and N is the size of the data set), we can say that we gain an improvement in overall deviation δD at the cost of a degradation in connectivity δC. For a number of clusters smaller than the true number K, we expect the ratio $R = \frac{\delta D}{\delta C}$ to be large: the separation of two clusters will trigger a great decrease in overall deviation, with only a small increase in connectivity. When we surpass the correct number of clusters this ratio will diminish: the decrease in overall deviation will be less significant but come at a high cost in terms of connectivity (because a true cluster is being split). Using this knowledge, let us consider a plot of the Pareto front. Due to the natural bias of both measures, the solutions are approximately ordered by the number of clusters they contain: K gradually increases from left to right. The distinct change in R occurring for the correct number of clusters can therefore be seen as a 'knee'. In order to help us correctly determine this knee, we can again use random reference data distributions. Clustering a number of such distributions using MOCK, provides us with a set of 'reference fronts' (see Figure 1.7).

Control data

The reference distributions are obtained using a Poisson model in eigen-space as suggested by Sarle [27]. Specifically, a principal component analysis is applied to the covariance matrix of the original data. The eigenvectors and eigenvalues obtained are then used for the definition of a uniform distribution: the data is generated within a hyperbox in eigenspace, where each side of the hyperbox is proportional in length to the size of the eigenvalue corresponding to this dimension. The resulting data is then back-transformed to the original data space.

Unfortunately, a normalization of the original 'solution front' using the 'reference fronts' is not as straightforward as the normalization of the performance curve for the Gap statistic. This is because both solution and control fronts contain not just

one, but a set of solutions for every value of K, and it is therefore not clear how individual points in the solution front should be normalized. We overcome this problem by a heuristic approach, described next.

Alignment of solution and control fronts

Given both solution and reference fronts, we set $K_{min} = 1$ and identify K_{max}, the highest number of clusters shared by all fronts. Subsequently, we restrict the analysis to solutions with a number of clusters $K \in [K_{min}, K_{max}]$. Solution points that are dominated by any reference point are also excluded from further consideration.

For each front, we then determine the minimum and maximum value of both overall deviation and connectivity, and use these to scale all objective values to lie within the region $[0, 1] \times [0, 1]$. We further transform the objective values by taking the square root of each objective, a step motivated by the observation that both overall deviation and connectivity show a non-linear development with respect to K. Overall deviation decreases very rapidly for the first few K, while changes for higher number of clusters are far less marked. Connectivity, in contrast rises very quickly for higher numbers of clusters, while initial changes in the degree of connectivity are rather small. This results in an uneven sampling of the range of objective values, which — in a plot of the Pareto front — shows in a high density of points at the tails, with fewer solution points in the centre. Taking the square root of the objective values is an attempt to reduce this 'squeezing' effect, and give a higher degree of emphasis to small (but distinct) changes in the objectives. By this means, the algorithm becomes more precise at identifying solutions situated in all parts of the Pareto front, in particular those at the tails which may correspond e.g. to partitionings with elongated cluster shapes or a high number of clusters.

Attainment scores

For both solution and reference fronts, we subsequently compute the attainment surfaces [11]. The attainment surface of a Pareto front is uniquely defined by the points in the front. It is the boundary in the objective space separating those points that are dominated by or equal to at least one of the data points in the front from those that no data point in the front dominates or equals (see Figure 1.7). For each point in the solution front we then compute its distance to the attainment surfaces of each of the reference fronts, and we refer to this distance as the 'attainment score'. For a given solution point p, we compute its attainment score as the Euclidean distance between p and the closest point on the the reference attainment surface.

Finally, we plot the attainment scores as a function of the number of clusters K (see Figure 1.7). All solutions corresponding to the local optima in the resulting plot are considered as promising solutions. The global maximum in this plot may be considered as a hypothesis as to the best solution.

1.3.3 Experimental setup

In this experimental section, we investigate the performance of MOCK's attainment method as a general tool for model selection. To make the investigation more meaningful, the performance of the attainment method is compared to an existing cluster validation technique, the Silhouette Width, which is highly-regarded in the clustering literature as a means to model selection.

To get a broad and detailed picture of how both methods perform, they are applied to the clustering results from four different clustering algorithms. Specifically, we run MOCK and three single-objective algorithms (the latter run for a range of different numbers of clusters $K \in [1, 50]$), and do this across the full range of data sets introduced in Section 1.2.4. On each data set, and for each algorithm, the two validation techniques are then used to perform model selection, that is, to select one or more partitionings from those generated by the clustering algorithm, as candidate 'best' solutions. The selected solutions are then objectively evaluated using the Adjusted Rand Index, an external measure of clustering quality. To isolate more clearly the performance of the validation techniques from the performance of the clustering algorithms, we also give the evaluation of the overall best solution returned by the clustering algorithm. In the following, we briefly describe the Silhouette Width (see also the Appendix) and state how the two validation techniques are applied to our data.

The Silhouette Width is a popular validation technique commonly used for model selection in cluster analyses. Notably, it can be considered a (fixed) nonlinear combination of two objectives, as it takes both intra-cluster distances and inter-cluster distances into account. The procedure for its use in model selection is as follows: for each $K \in [1, 50]$, the Silhouette Width of the corresponding clustering solution is computed. In a plot of the Silhouette Width as a function of the number of clusters, the global maximum and all local maxima are identified. The global maximum is considered the estimated best solution. All local maxima are considered as alternative, potentially interesting solutions.

Similarly, MOCK's attainment method can be adapted for model selection on the clustering results of any single-objective algorithm. For this purpose, clustering results for a range of $K \in [1, 50]$ are obtained both on the original data set and random control data (which is generated in the same way as described in Section 1.3.2). The resulting sequences are then subjected to the procedure detailed in Section 1.3.2, in order to obtain attainment scores for all solutions. In a plot of the attainment score as a function of the number of clusters, the global maximum and all local maxima are identified. The global maximum is considered the guess for the best solution. All local maxima are considered as alternative, potentially interesting solutions.

1.3.4 Experimental results

The Silhouette Width assumes compact, spherically shaped clusters. We would therefore expect it to perform well for data sets exhibiting this type of structure, but to perform poorly for data containing arbitrarily shaped, elongated clusters. In contrast,

we would expect MOCK's attainment method to perform more uniformly well for data of different structures.

Table 1.2, Table 1.3, Table 1.4 and Table 1.5 shows the results obtained for model selection on the results returned by the four different algorithms. We have seen before, that MOCK is the only one of the three algorithms that perform robustly across the entire range of data sets. The results in Table 1.2 indicate that this advantage can be largely maintained when using the attainment score for model selection, but is strongly reduced, if the Silhouette Width is used. While the Silhouette Width performs (as expected) very well for model selection on the Square and the Sizes data sets, its performance breaks down drastically for those data with elongated cluster shapes.

In Table 1.3, no advantage can be observed for MOCK's attainment score method. This is intuitively plausible, as the K-means algorithm only generates spherically shaped, compact clusters. All of the partitionings generated by the algorithm are therefore in concordance with the assumptions made by the Silhouette Width, and the Silhouette Width can therefore be expected to perform well. In such a scenario, the additional flexibility of the attainment score is not rewarded, but instead results in the introduction of noise.

In Table 1.4 and Table 1.5 a slight advantage of the attainment method can be observed. In general, it seems to cope better with the data sets containing elongated data sets, and, for single link, its estimate of the best clustering solution is generally of much higher quality.

Overall, it seems that the utility of multiobjective cluster validation strongly depends on the algorithm used for the generation of the clustering solutions. In the context of multiobjective clustering, where the full extent of the true Pareto front is approximated, a clear advantage to the multiobjective approach can be observed. In contrast, for traditional clustering algorithms such as K-means, which are based on very specific cluster models, the use of an analogous validation technique based on identical assumptions may be preferable.

1.4 Conclusion

In this chapter we have identified two subtasks in clustering, in which the use of a multiobjective framework may be beneficial: (1) the actual process of clustering, that is, the generation of clustering solutions; and (2) model selection, that is, the selection of the best clustering out of a set of given partitionings. Both processes require the definition of a clustering criterion, and the algorithms' capability to deal with different data properties highly depends on this choice. We have argued that that the simultaneous consideration of several objectives may be a way to obtain a more consistent performance (across a range of data properties) for both subtasks. This idea has been illustrated through the development of a multiobjective clustering algorithm and a multiobjective scheme for model selection. The resulting algorithms have been evaluated across a range of different data sets, and results have been com-

Table 1.2. Results for MOCK. The overall best solution found (in terms of the Adjusted Rand Index), and the solutions identified using the Silhouette Width and the attainment score are evaluated using the Adjusted Rand Index (Rand) and the number of clusters (K). Here, Max A-score' and 'Max S-score' refer to the solution corresponding to the global maximum in the plot of the attainment score and the Silhouette Width, respectively. 'Best local A-score' and 'Best local S-score' refer to the best (in terms of the Adjusted Rand Index) out of the set of solutions corresponding to local maxima in the plot of the attainment score and the Silhouette Width, respectively. Values presented are averages over 50 sample data sets.

Problem	Best Solution		Max A-score		Max S-score		Best local A-score		Best local S-score	
	K	Rand	K	Rand	K	Rand	K	Rand	K	Rand
Square1	4.12	0.96342	4.02	0.95919	4	**0.96184**	4.02	0.95919	4	**0.96184**
Square2	4.2	0.92661	4.12	0.91546	4	**0.92438**	4.24	0.91790	4	**0.92438**
Square3	4.42	0.86767	4.08	0.85653	4	**0.86251**	4.08	0.85653	4	**0.86251**
Sizes1	4.24	0.96465	4.02	0.96113	4	**0.96314**	4.02	0.96113	4	**0.96314**
Sizes2	4.16	0.97110	4.06	0.96724	4	**0.96868**	4.1	0.96727	4	**0.96868**
Sizes3	4.14	0.97575	3.92	0.9411	4	**0.97394**	4.04	0.97109	4	**0.97394**
Long1	2	0.99984	2.02	**0.99745**	7.68	0.32103	2.02	**0.99745**	6.48	0.36782
Long2	2	0.99992	2.04	**0.99748**	7.76	0.31094	2.04	**0.99748**	6.44	0.37415
Long3	2.02	0.99965	2.44	**0.93692**	7	0.21439	2.2	**0.96878**	3.72	0.38018
Smile1	4	1	4	**1**	14.24	0.72711	4	**1**	10.56	0.77839
Smile2	4	1	4	**1**	13	0.55059	4	**1**	10.5	0.63436
Smile3	4	1	4	**1**	15.96	0.32214	4	**1**	11.48	0.39329
Triangle1	4	1	4	**1**	4.02	0.99680	4	**1**	4.02	0.99680
Triangle2	4.04	0.98652	4.04	**0.97692**	4	0.97165	4.04	**0.97692**	4	0.97165

Table 1.3. Results for K-means. The overall best solution found (in terms of the Adjusted Rand Index), and the solutions identified using the Silhouette Width and the attainment score are evaluated using the Adjusted Rand Index (Rand) and the number of clusters (K). Here, Max A-score' and 'Max S-score' refer to the solution corresponding to the global maximum in the plot of the attainment score and the Silhouette Width, respectively. 'Best local A-score' and 'Best local S-score' refer to the best (in terms of the Adjusted Rand Index) out of the set of solutions corresponding to local maxima in the plot of the attainment score and the Silhouette Width, respectively. Values presented are averages over 50 sample data sets.

Problem	Best Solution		Max A-score		Max S-score		Best local A-score		Best local S-score	
	K	Rand	K	Rand	K	Rand	K	Rand	K	Rand
Square1	4	0.96500	4	**0.96500**	4	**0.96500**	4	**0.96500**	4	**0.96500**
Square2	4	0.935684	4	**0.93568**	4	**0.93568**	4	**0.93568**	4	**0.93568**
Square3	4	0.88432	4.02	0.88259	4	**0.88432**	4.02	0.88259	4	**0.88432**
Sizes1	4	0.96648	4	**0.96648**	4	**0.96648**	4	**0.96648**	4	**0.96648**
Sizes2	4	0.96532	4	**0.96532**	4	**0.96532**	4	**0.96532**	4	**0.96532**
Sizes3	4	0.96336	4	**0.96336**	4	**0.96336**	4	**0.96336**	4	**0.96336**
Long1	5.04	0.35500	12.04	0.20836	8.42	**0.27396**	8.38	0.27602	7.62	**0.29356**
Long2	4.92	0.33803	12.48	0.18235	8.32	**0.25925**	8.78	0.24886	7.02	**0.29083**
Long3	4.46	0.21694	13.86	0.08793	6.64	0.18584	11.36	0.11241	4.54	0.21598
Smile1	30.6	0.74397	9.66	0.65296	41.58	**0.73858**	14.8	0.69363	30.18	**0.74356**
Smile2	25.22	0.55347	12.58	0.53044	35.2	**0.53460**	7.64	0.53681	24.94	**0.55340**
Smile3	34.02	0.34886	13.1	0.29340	43.02	**0.33693**	13.84	0.29585	32.28	**0.34815**
Triangle1	4	0.95805	5.34	0.90414	4	**0.95805**	5.3	0.90565	4	**0.95805**
Triangle2	4	0.88575	5.16	0.77339	4	**0.88575**	5.12	0.77700	4	**0.88575**

Table 1.4. Results for average link. The overall best solution found (in terms of the Adjusted Rand Index), and the solutions identified using the Silhouette Width and the attainment score are evaluated using the Adjusted Rand Index (Rand) and the number of clusters (K). Here, Max A-score' and 'Max S-score' refer to the solution corresponding to the global maximum in the plot of the attainment score and the Silhouette Width, respectively. 'Best local A-score' and 'Best local S-score' refer to the best (in terms of the Adjusted Rand Index) out of the set of solutions corresponding to local maxima in the plot of the attainment score and the Silhouette Width, respectively. Values presented are averages over 50 sample data sets.

Problem	Best Solution		Max A-score		Max S-score		Best local A-score		Best local S-score	
	K	Rand	K	Rand	K	Rand	K	Rand	K	Rand
Square1	4.16	0.93701	4	**0.93671**	4	**0.93671**	4	**0.93671**	4	**0.93671**
Square2	4.24	0.89713	4	**0.89661**	4	**0.89661**	4.1	**0.89664**	4	0.89661
Square3	4.12	0.82932	4.02	0.82147	4.06	**0.82921**	4.06	**0.82921**	4.06	**0.82921**
Sizes1	4.32	0.94117	4	**0.94079**	4	**0.94079**	4	**0.94079**	4	**0.94079**
Sizes2	4.28	0.95103	4	**0.95089**	4	**0.95089**	4	**0.95089**	4	**0.95089**
Sizes3	4.26	0.96008	4	**0.95990**	4	**0.95990**	4.08	**0.95990**	4	0.95990
Long1	7.64	0.47172	14.38	**0.21911**	7.78	0.17631	11.92	0.28632	9.3	**0.37684**
Long2	7.84	0.45719	15.4	**0.18893**	6.7	0.12544	12.94	0.24133	9.3	**0.36842**
Long3	7.32	0.37855	16.16	**0.08473**	5.84	0.082558	15.98	0.09140	10.24	**0.25754**
Smile1	11.98	0.80083	8.9	0.68408	17.42	**0.72330**	11.78	**0.74898**	17.1	0.72458
Smile2	10.54	0.75607	4.82	**0.55600**	18.62	0.52529	11.08	**0.69273**	16.64	0.54517
Smile3	11.22	0.38256	10.28	0.30292	18	**0.30426**	12.2	**0.37947**	17.1	0.31562
Triangle1	4.82	0.99259	4.8	**0.9886**	4	0.98493	4.8	**0.98863**	4	0.98493
Triangle2	5.02	0.954345	4.64	0.935819	4.06	**0.94234**	4.58	**0.94283**	4.06	0.94234

Table 1.5. Results for single link. The overall best solution found (in terms of the Adjusted Rand Index), and the solutions identified using the Silhouette Width and the attainment score are evaluated using the Adjusted Rand Index (Rand) and the number of clusters (K). Here, Max A-score' and 'Max S-score' refer to the solution corresponding to the global maximum in the plot of the attainment score and the Silhouette Width, respectively. 'Best local A-score' and 'Best local S-score' refer to the best (in terms of the Adjusted Rand Index) out of the set of solutions corresponding to local maxima in the plot of the attainment score and the Silhouette Width, respectively. Values presented are averages over 50 sample data sets.

Problem	Best Solution		Max A-score		Max S-score		Best local A-score		Best local S-score	
	K	Rand	K	Rand	K	Rand	K	Rand	K	Rand
Square1	37.76	0.82555	14.54	**0.32341**	2.72	0.03742	14.24	0.33951	38.3	**0.82400**
Square2	40.12	0.45803	11.56	**0.08107**	2	8.02944e-07	10.46	0.08108	40.88	**0.45713**
Square3	41.32	0.10803	11.84	**0.00568**	2	1.07145e-06	12.36	0.00570	43.08	**0.10770**
Sizes1	36.22	0.84291	14.4	**0.38400**	3.06	0.06278	15.5	0.40410	36.28	**0.84184**
Sizes2	35.64	0.82880	13.72	**0.42990**	2.18	0.02102	15.92	0.47064	35.38	**0.82803**
Sizes3	34.94	0.87194	12.58	**0.49804**	2.7	0.042279	15.04	0.53626	35.74	**0.87094**
Long1	3.48	0.99570	4.42	**0.97150**	2.02	0.30000	4.32	0.97182	8.7	**0.97394**
Long2	3.42	0.99595	4.58	**0.99073**	2.06	0.20012	4	**0.99366**	9.52	0.97057
Long3	3.32	0.99612	4.98	**0.98881**	2.06	0.26364	4.46	**0.99138**	7.58	0.97434
Smile1	4	1	4	**1**	4	**1**	4	**1**	4	**1**
Smile2	4	1	4.04	0.99992	4	**1**	4.04	0.99992	4	**1**
Smile3	4	1	4.08	**0.99991**	3.6	0.88220	4.08	0.99991	4	**1**
Triangle1	4.14	0.99979	4.14	**0.99979**	4.14	**0.99979**	4.14	**0.99979**	4.14	**0.99979**
Triangle2	24.4	0.94976	11.12	**0.63951**	5.46	0.29776	13.7	0.72440	24.68	**0.94832**

pared to those obtained using traditional single-objective approaches. Experimental results indicate some general advantages to the multiobjective approach.

Acknowledgements

Julia Handl gratefully acknowledges support of a scholarship from the Gottlieb Daimler- and Karl Benz-Foundation, Germany. Joshua Knowles is supported by a David Phillips Fellowship from the Biotechnology and Biological Sciences Research Council (BBSRC), UK.

References

1. M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering points to identify clustering structure. In *Proceedings of the 1999 International Conference on Management of Data*, pages 49–60. ACM Press, 1999.
2. S. Bandyopadhyay and U. Manlik. Nonparametric genetic clustering: comparison of validity indices. *IEEE Transactions on Systems, Man and Cybernetics*, 31:120–125, 2001.
3. J. Bilmes, A. Vahdat, W. Hsu, and E.-J. Im. Empirical observations of probabilistic heuristics for the clustering problem. Technical Report TR-97-018, International Computer Science Institute, University of California, Berkeley, CA, 1997.
4. D. W. Corne, J. D. Knowles, and M. J. Oates. The pareto envelope-based selection algorithm for multiobjectice optimization. In *Proceedings of the Fifth Conference on Parallel Problem Solving from Nature*, pages 839–848, 2000.
5. D. W. Corne, 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. David W. Corne, Nick R. Jerram, Joshua D. Knowles, and Martin J. Oates. PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 283–290. Morgan Kaufmann Publishers, 2001.
7. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification, second edition*. John Wiley and Son Ltd, 2001.
8. M. Ester, H. P. Kriegel, and J. Sander. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data-Mining*. AIII Press, 1996.
9. V. Estivill-Castro. Why so many clustering algorithms: A position paper. *ACM SIGKDD Explorations Newsletter Archive*, 4:65–75, 2002.
10. B. S. Everitt. *Cluster analysis*. Edward Arnold, 1993.
11. C. M. Fonseca and P. J. Fleming. On the performance assessment and comparison of stochastic multiobjective optimizers. In *Proceedings of the Fourth International Conference onParallel Problem Solving from Nature*, pages 584–593. Springer-Verlag,, 1996.
12. J. Handl and J. Knowles. Evolutionary multiobjective clustering. In *Proceedings of the Eighth International Conference on Parallel Problem Solving from Nature*, pages 1081–1091. Springer-Verlag, 2004.
13. J. Handl and J. Knowles. Multiobjective clustering with automatic determination of the number of clusters. Technical Report TR-COMPSYSBIO-2004-02, UMIST, Manchester, UK, 2004.

14. J. Handl and J. Knowles. Exploiting the trade-off: the benefits of multiple objectives in data clustering. In *Proceedings of the Third International Conference on Evolutionary Multicriterion Optimization*, pages 547–560. Springer-Verlag, 2005.
15. J. Handl and J. Knowles. Improvements to the scalability of multiobjective clustering. In *IEEE Congress on Evolutionary Computation*. IEEE Press, 2005. To appear.
16. T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer-Verlag, 2001.
17. A. Hubert. Comparing partitions. *Journal of Classification*, 2:193–198, 1985.
18. J. Knowles J. Handl and D. B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 2005. Epub ahead of print. Available from http://bioinformatics.oupjournals.org/cgi/reprint/bti517v1.
19. A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31:264–323, 1999.
20. J. Kleinberg. An impossibility theorem for clustering. In *Proceedings of the 15th Conference on Neural Information Processing Systems*. The Internet, 2002.
21. L. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967.
22. G. McLachlan and T. Krishman. *The EM algorithm and extensions*. John Wiley and Son Ltd, 1997.
23. Y.-J. Park and M.-S. Song. A genetic algorithm for clustering problems. In *Proceedings of the Third Annual Conference on Genetic Programming*, pages 568–575, Madison, WI, 1998. Morgan Kaufmann.
24. J. M. Pena, J. A. Lozana, and P. Larranaga. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Letters*, 20:1027–1040, 1999.
25. V. J. Rayward-Smith, I. H. Osman, C. R. Reeves, and G. D. Smith. *Modern heuristic search methods*. John Wiley and Son Ltd, 1996.
26. 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.
27. W. S. Sarle. Cubic clustering criterion. Technical report, SAS Technical Report A-108, Cary, NC: SAS Institute Inc, 1983.
28. A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research*, 3:583–617, 2002.
29. G. Syswerda. Uniform crossover in genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 2–9. Morgan Kaufmann Publishers, 1989.
30. R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the Gap statistic. Technical report, Stanford University, 2000.
31. A. Topchy, A. K. Jain, and W. Punch. Clustering ensembles: Models of consensus and weak partitions. Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004.
32. E. Vorhees. *The effectiveness and efficiency of agglomerative hierarchical clustering in document retrieval*. PhD thesis, Department of Computer Science, Cornell University, 1985.
33. Darrell Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4:65–85, 1994.
34. Robin J. Wilson and John J. Watkins. *Graphs: an introductory approach: a first course in discrete mathematics*. John Wiley and Sons, 1990.

1.5 Appendix

1.5.1 Evaluation functions

Silhouette Width

The Silhouette Width [26] for a partitioning is computed as the average Silhouette value over all data items. The Silhouette value for an individual data item i, which reflects the confidence in this particular cluster assignment, is computed as

$$S(i) = \frac{b_i - a_i}{\max(b_i, a_i)},$$

where a_i denotes the average distance between i and all data items in the same cluster, and b_i denotes the average distance between i and all data items in the closest other cluster (which is defined as the one yielding the minimal b_i).

The Silhouette Width return values in the interval $[-1, 1]$ and is to be maximized.

Adjusted Rand Index

In all experiments, clustering quality is objectively evaluated using the Adjusted Rand Index, an external measure of clustering quality, which is is a generalization of the Rand Index.

The Rand indices are based on counting the number of pair-wise co-assignments of data items. The Adjusted Rand Index additionally introduces a statistically induced normalization in order to yield values close to 0 for random partitions. Using a representation based on contingency tables, the Adjusted Rand Index [17] is given as

$$R(U,V) = \frac{\sum_{lk}\binom{n_{lk}}{2} - \left[\sum_l \binom{n_{l\cdot}}{2} \cdot \sum_k \binom{n_{\cdot k}}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_l \binom{n_{l\cdot}}{2} + \sum_k \binom{n_{\cdot k}}{2}\right] - \left[\sum_l \binom{n_{l\cdot}}{2} \cdot \sum_k \binom{n_{\cdot k}}{2}\right] / \binom{n}{2}},$$

where n_{lk} denotes the number of data items that have been assigned to both cluster l and cluster k.

The Adjusted Rand Index return values in the interval $[0, 1]$ and is to be maximized.

1.5.2 Algorithms

K-means

Starting from a random partitioning, the K-means algorithm repeatedly (i) computes the current cluster centres (that is, the average vector of each cluster in data space) and (ii) reassigns each data item to the cluster whose centre is closest to it. It terminates when no more reassignments take place. By this means, the intra-cluster

variance, that is, the sum of squares of the differences between data items and their associated cluster centres, is locally minimized.

Our implementation of the K-means algorithm is based on the batch version of K-means, that is, cluster centres are only recomputed after the reassignment of all data items. To reduce suboptimal solutions K-means is run repeatedly (100 times) using random initialisation (which is known to be an effective initialization method [24]) and only the best result in terms of intra-cluster variance is returned.

Hierarchical clustering

In general, agglomerative clustering algorithms start with the finest partitioning possible (that is, singletons) and, in each iteration, merge the two least distant clusters. They terminate when the target number of clusters has been obtained. Alternatively, the entire dendrogram can be generated and be cut at a later point.

Single link and average link agglomerative clustering only differ in the linkage metric used. For the linkage metric of average link, the distance between two clusters C_i and C_j is computed as the average dissimilarity between all possible pairs of data elements i and j with $i \in C_i$ and $j \in C_j$. For the linkage metric of single link, the distance between two clusters C_i and C_j is computed as the smallest dissimilarity between all possible pairs of data elements i and j with $i \in C_i$ and $j \in C_j$.

Parameter settings for MOCK

Parameter settings for MOCK are given in Table 1.6 and are kept constant over all experiments.

Table 1.6. Parameter settings for MOCK, where N is data set size.

Parameter	*setting*
Number of generations	500
External population size	1000
Internal population size	10
#(Initial solutions) $fsize$	100
Initialization	Minimum spanning tree and K-means ($L = 20$)
Mutation type	L nearest neighbours ($L = 20$)
Mutation rate p_m	$p_m = \frac{1}{N} + (\frac{l}{N})^2$
Recombination	Uniform crossover
Recombination rate p_c	0.7
Objective functions	Overall deviation and connectivity ($L = 20$)
#(Reference distributions)	3