COMPARING INITIALISATION METHODS FOR THE HEURISTIC MEMETIC CLUSTERING ALGORITHM

B.G.W. Craenen[*†], *T. Ristaniemi*[*], *A.K. Nandi*[†*]

[*]Department of Mathematical Information Technology, University of Jyväskylä, Jyväskylä, Finland
[†]School of Engineering & Design, Brunel University, London, UK

ABSTRACT

In this study we investigate the effect five initialisation methods from literature have on the performance of the Heuristic Memetic Clustering Algorithm (HMCA). The evaluation is based on an extensive experimental comparison on three benchmark datasets between HMCA and the commonly-used k-Medoids algorithm. Analysis of the experimental effectiveness and efficiency metrics confirms that the HMCA substantially outperforms k-Medoids, with the HMCA capable of finding bestter clusterings using substantially less computation effort. The *Sample* and *Cluster* initialisation methods were found to be the most suitable for the HMCA, with the results of the k-Medoids suggesting this to be the case for other algorithms as well.

Index Terms— Machine Learning, Memetic Algorithms, Clustering, Heuristics

1. INTRODUCTION

Clustering is a data and signal processing task with the objective to determine a finite set of categories, called clusters, to describe a dataset according to the similarities among its data objects [1, 2]. Applications of clustering are many and range from market segmentation [3] and image processing [4], to document categorisation and web mining [5]. Recently clustering has gained prominence for signal processing in the field of bioinformatics [6–9].

Clustering can then be defined as follows. A dataset, $\mathbf{X}$, has N data objects: $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, with each data object, $\mathbf{x}$, having n features or attributes: $\mathbf{x} = \{x^1, \ldots, x^n\}$. A configuration of clusters, or clustering, $\mathbf{C}$, has k clusters: $\mathbf{C} = \{\mathbf{c}_1, \ldots \mathbf{c}_k\}$, with each cluster, $\mathbf{c}$, a subset of data objects from the dataset: $\mathbf{c} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathbf{c}|}\}$, with $\mathbf{x}_i \in \mathbf{X}$. Clusters are not allowed to be empty, $\mathbf{c}_i \neq \emptyset$, or overlap, $\mathbf{c}_1 \cup \mathbf{c}_2 \cup \ldots \cup \mathbf{c}_k = \mathbf{X}$ and $\mathbf{c}_i \cap \mathbf{c}_j \neq \emptyset$ for $i \neq j$.

Clustering algorithms seek to assign all data objects of a dataset to clusters and, from an optimisation perspective, strive to maximise the homogeneity of data objects within a cluster, while simultaneously maximising the heterogeneity between clusters [4, 10]. Measuring similarity (homogeneity) is tackled indirectly by using a distance measure quantifying the degree of dissimilarity (heterogeneity) between data objects. Usually distance measures are defined so that similar data objects have lower dissimilarity values, while dissimilar objects have higher values [4, 10].

In Machine Learning, clustering is deemed to be one of the most difficult and challenging problems, mostly due to its unsupervised nature, and the implication that the structural characteristics of the dataset remain unknown [11]. Clustering is formally considered a particular kind of NP-hard grouping problem [12]. This has stimulated the search for efficient approximation algorithms, especially, for more general meta-heuristics like Evolutionary Algorithms (EAs), and Memetic Algorithms (MAs), capable of evolving near-optimal solutions in reasonable time. Many clustering EAs and MAs have been proposed in literature (see [13]).

This study investigates the effect five initialisation methods from literate have on the performance of the Heuristic Memetic Clustering Algorithm (HMCA), as well as the relative effect this has when compared with the k-Medoids. In [14] we designed the HMCA as an easy-to-understand yet powerful clustering algorithm, adaptable for a wide variety of clustering problems and dataset. The basis for this adaptability lies in its single local-search operator, incorporated into the MA meta-heuristic. By applying this operator with different heuristics, the HMCA can be tailored to a wide variety of clustering problems and datasets. In [14], the performance of the HMCA was compared with the commonly used k-Medoids algorithm on three benchmark datasets. The HMCA was found to be substantially more effective in finding good clusterings, and efficient in being able to do so using fewer computational resources.

For all its demonstrated effectiveness and efficiency in clustering datasets, further analysis of the experimental results showed that HMCA's initial clusterings were often of substantially worse quality than those available to k-Medoids. Subsequently the HMCA had to expend significant computational effort catching up. The HMCA being stochastic, its ability to invariably do so accentuates its power and perfor-

This work was financially supported by TEKES (Finland) under grant 40334/10 "Machine Learning for Future Music and Learning Technologies" (MUSCLES). Asoke K. Nandi would like to thank TEKES for the award of the Finland Distinguished Professorship

mance. But it also lead to the question we seek to answer in this study: can the performance of the HMCA be further improved by using other, more suited, initialisation methods?

To answer this research question, we present five commonly used initialisation methods from literature [15] and apply these initialisation methods to both the HMCA and k-Medoids, allowing a fair comparison between the two. We then use an experimental layout similar to the one used in [14] to compare the efficiency and effectiveness of both algorithms on three datasets, varying the initialisation methods for both.

The remainder of this study is then organised as follows. Section 2 provides a summary description of the HMCA. In Section 3 we describe the initialisation methods. Section 4 the describes the experimental setup, while Section 5 presents the experimental results. Finally, Section 6 provides some discussion and a conclusion.

2. ALGORITHM

The HMCA was first presented in [14] and we refer to that publication for a detailed description of the algorithm. Because of the limited space available, we provide only a summary description of the HMCA, sufficient for understanding the issues raised here.

The HMCA is a MA for clustering datasets. MAs are a subset of EAs, in that they resemble EAs but include a local-search component. MAs are sometimes referred to as Baldwinian EAs, Lamarckian EAs, Cultural Algorithms, or Genetic Local-Search Algorithms [16]. The HMCA's local-search component is a single local-search operator using heuristics to guide the evolutionary process towards better clusterings.

The HMCA evolutionary process closely follows the canonical EA process: Clusterings are encoded in individuals, and an initial population of individuals is initialised using an initialisation method. Each individual has a value quantifying its quality or fitness, assigned by the fitness function. The HMCA fitness function calculates the cumulative inner-distance of a clustering:

$$f(\mathbf{C}) = \sum_{\mathbf{c}_i \in \mathbf{C}} \sum_{l=1}^{|\mathbf{c}_i|-1} \sum_{m=l+1}^{|\mathbf{c}_i|} d(x_l, x_m) \qquad (1)$$

with $d(x_l, x_m)$ the Euclidean distance between data object x_l and x_m as a distance metric. The fitness value is to be minimised, i.e., good clusterings have tight clusters.

The HMCA then iteratively approximates towards global optimal clusterings by applying a single local-search operator to each individual in the (parent) population. This operator generates one child individual for each parent individual by reassigning one data object from one cluster to another. Empty clusters are avoided, but when they occur are repaired by moving a random data object from the biggest cluster to the empty cluster (iteratively if there is more than one empty cluster). The (repaired) offspring individuals are then assigned a fitness value, and placed in the offspring population. Both parent and offspring populations are merged and duplicate clusterings (based on cluster content) are discarded. Sorted by fitness values, the merged population is pruned to a size equal to the population size parameter by discarding the worst individuals, thus creating the parent population for the next iteration of the HMCA.

The iterative process is halted when the HMCA has exhausted a set maximum number of distance calculations. Distance calculations are used by the fitness function, but also by the local-search operator, and by the initialisation method.

The innovative feature of the HMCA is its heuristic local-search operator. It uses up to three heuristics to select a data object, and then to select a cluster to subsequently assign it to. Selecting the data object may involve two heuristics: one to first select a cluster, from which another then selects a data object, or, alternatively, just one selecting a data object from all data objects in the clustering without regard to which cluster it is assigned. The three heuristic types are then: one for selecting a (source) *cluster*, one for selecting a *data object*, and one for selecting *label* (or destination cluster).

All three heuristic types have a benchmark or reference heuristic, called a *null-heuristic*. These are, respectively: select no cluster (data objects are selected from all available), select a random data object, and select a random label. Four heuristics are defined for both the cluster and data object heuristic types, while six are defined for the label heuristic type. These heuristics are defined around distance metrics, and use distance calculations, with selection either deterministic or proportionally random. We refer to [14] for precise definitions of all heuristics. Including the null-heuristics, the HMCA uses a total of five cluster and data object heuristics, and seven label heuristics.

The HMCA then has the following input parameters: a dataset to cluster, the number of clusters to cluster it in, the size of the population to use, the maximum number of distance calculations, and the three heuristics used by the local-search operator. A HMCA *variant* for a dataset is be defined by the three heuristics used by the local-search operator. Some HMCA variants are expected to have superior performance than others, and differing performance for different datasets.

3. INITIALISATION METHODS

Originally, in [14], the HMCA used a two-step approach for initialising individuals: first, all clusters were assigned a unique, randomly chosen data object; then all remaining data objects were assigned to random clusters. This initialisation methods avoids initial individuals having empty clusters, but the random assignment of the remaining data objects is also likely to result in unfavourable initial partitions. Mostly be-

cause the resulting clusters are likely to be mixed up to an unfavourably high degree.

The likelihood of producing unfavourably, mixed-up initial clusters was already mentioned in [13], which then goes on to state that this does constitute an effective approach for testing algorithms against hard evaluation scenarios. The authors agree, but this seems rather of limited interest to those simply wishing to evolve good quality clusterings. Furthermore, This drawback makes for rather unfair comparisons between EAs (and MAs) and other algorithm types who need not suffer from this. So, while it is comforting to the find that the HMCA was still capable of comparatively superior performance in [14], in this study, similar to [15], we investigate the (comparative) performance of the HMCA when the original initialisation method is replaced with one of the following five methods:

Sample Like the original initialisation method, *Sample* first assigns each cluster a unique randomly selected data object as a medoid, but then assigns the remaining data objects to the cluster with the least distance between itself and the medoid.

Cluster This initialisation method takes a 10% randomly selected sample of all data objects, and then uses a single iteration of k-Medoids to find k medoids among them. These are assigned to each cluster, with the remaining data objects assigned as with *Sample*.

Uniform This method calculates the value ranges of each attribute by taking the minimum and maximum values for each attribute for each data object in the dataset. k new data objects are then generated by taking a random value from a uniform random distribution between these ranges for each attribute. These new data objects are then matched to data objects in the dataset by selecting the one closest to it, and used as medoids for each cluster. The remaining data objects are assigned as with *Sample*.

KA0* and *KA1 These two initialisation methods are based on the Kaufman Approach (KA) [1]. A deterministic constructive approach, this initialisation method essentially compares the pair-wise distance between all data objects, subtracting from it the minimum distance to the data object and already selected medoids. Using this measure, the KA iteratively selects k medoids that maximise the heterogeneity in the dataset's hyperplane. The difference between *KA0* and *KA1* reflects an interpretation difference in the description in [1]. *KA0* uses accumulative distances between data objects, while *KA1* uses single distances. The remaining data objects are assigned as with *Sample*.

4. EXPERIMENTAL SETUP

The experimental setup used here is similar to the one used in [14]. The initialisation methods described in Section 3 were implemented for use in both the HMCA and k-Medoids. The k-Medoids algorithm was further adapted to use the same performance measures as the HMCA, making a direct comparison both possible and fair.

Both the effectiveness and efficiency of both algorithms are measured. Effectiveness is an expression of the ability of the algorithms to find (evolve) good quality clusterings of the dataset. Clustering quality in the HMCA is assessed by the fitness value, i.e., the cumulative inner-cluster distance. The same measure is used by k-Medoids, but, since maintaining fitness values is not integral to k-Medoids, the distance calculations required for calculating it are not included when measuring efficiency.

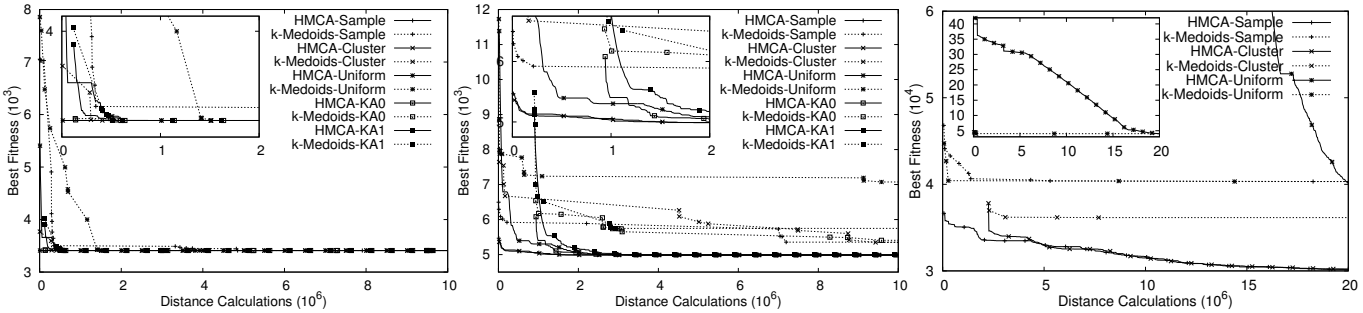
Efficiency is an expression of the amount of computational effort required by the algorithm to find good clusterings. Since the commonly used *wallclock time to solution* metric is affected by range of unquantifiable factors (such as computing platform, implementation quality, etc.), as in [14], we use the number of distance calculations as the atomic efficiency measure instead.

As in [14], we use three commonly-used datasets to evaluate the performance of both algorithms: the Iris flower dataset (Iris), the Glass Identification dataset (Glass), and a generated Quadrature Amplitude Modulation dataset (16-QAM). The Iris dataset is a collected multivariate dataset of three ($k = 3$) Iris flowers for a total of $N = 150$ data objects with $n = 4$ measured features (attributes) each [17]. The Glass dataset is a multivariate dataset consisting of $n = 9$ collected chemical compositions of glass, for a total of $N = 217$ data objects, clustered into $k = 6$ clusters of variable size [18]. The 16-QAM dataset is a generated dataset consisting of a $N = 1024$ data object signal stream, modulated randomly on a $4 \times 4 = 16$ two dimensional rectangular grid ($k = 16$), giving each data object $n = 2$ features (coordinates). After generation of the signal stream, 12 dB of Gaussian noise was added. With different numbers of clusters to be found, with different degrees of overlap between those clusters, and varying numbers of data objects and features, we believe that together these datasets provide a diverse environment to evaluate and compare both algorithms on.

Parameters for the algorithms then include: the number of clusters ($k = 3$ for Iris, $k = 6$ for Glass, and $k = 16$ for 16-QAM), population size (for HMCA: $\{1, 2, 5, 10\}$), and all combinations of the heuristics for HMCA ($5 \cdot 5 \cdot 7 = 175$ HMCA variants). Both algorithms are stochastic, and were repeatedly run 25 times to gain sufficient statistical accuracy. They were both given $10,000,000$ distance calculations to cluster the Iris and Glass datasets, and $20,000,000$ distance calculations for the 16-QAM dataset. Preliminary experimentation showed that these values allowed both algorithms sufficient computational effort for a fair comparison.

5. RESULTS

Both the effectiveness and efficiency of both algorithms can be assessed by examining the fitness value per distance cal-



(a) Iris (b) Glass (c) 16-QAM

Fig. 1. Experimental results of both HMCA and k-Medoids, for datasets Iris (a), Glass (b), and 16-QAM (c).

culations graph. Effectiveness can be compared by looking at the eventual fitness value of the found clusterings, while efficiency can be assessed by looking at the behaviour of the fitness values during the run. The results of the experiments are show in Figure 1, which includes a graph for each of the three dataset (Iris (a), Glass (b), and 16-QAM (c)). Each graph includes a curve for both algorithms, for each initialisation method, for a total of ten curves (except 16-QAM, see below). The curves for the HMCA results are depicted using solid lines, while dotted lines were used for k-Medoid's results. The horizontal and vertical axis measure the number of distance calculations and the best fitness value, respectively. Each curve represents the best performance behaviour out of 25 runs, with the HMCA results represented by compound curves over all HMCA variants (see [14]). Each subfigure includes an inset graph focusing on early performance behaviour for the Iris and Glass datasets, and the *Uniform* results for the 16-QAM dataset (see below).

Subfigure (a) shows both algorithms capable of quickly finding a local optimal clustering for the Iris dataset, with similar eventual effectiveness. The HMCA, however, shows superior efficiency, descending towards the local optimal clustering faster, often substantially so. The inset shows that the *Sample* and *Cluster* initialisation methods proved to be the most suited for the HMCA, with both curves almost on top of each other, and little further improvement (needed) after initialisation. The *KA0* initialisation method had the best performance for k-Medoids, although the difference is relatively small.

A substantial effectiveness difference between the HMCA and k-Medoids was found for the Glass dataset, with the HMCA consistently finding better clusterings, for all five initialisation methods. The HMCA was also substantially more efficient than k-Medoids, with subfigure (b) indicating that the HMCA never needs more that three million distance calculations to find a good clustering, with k-Medoids incapable to do so even when using all ten million distance calculations available. In the inset we found, again, that the *Sample* and *Cluster* initialisation methods gave the HMCA the best performance, with the *Sample* method providing the best performance for k-Medoids.

There are only six curves out of ten in subfigure (c), with the *KA0* and *KA1* curves not depicted. This is because both *KA0* and *KA1* are expensive initialisation methods, and become exponentially so as the number of data objects and the number of clusters in the dataset increases. For the 16-QAM dataset, both initialisation methods use so many distance calculations that they used up more distance calculations than the maximum available.

The remaining curves in subfigure (c) show similar performance behaviour for the 16-QAM dataset for both algorithms, to what was seen in subfigure (b). We find k-Medoids stuck in a worse local optimal clustering, with a substantial efficiency difference between the two algorithm as well. The HMCA outperforms k-Medoids substantially for both the *Sample* and *Cluster* initialisation methods, with these initialisation methods providing the best performance for both algorithms.

The exception in subfigure (c) is the HMCA performance when using the *Uniform* initialisation method, shown in the inset. The poor performance of this method is caused by the way this method randomly selects medoid coordinates in the 16-QAM two-dimensional feature space. The generation and selection procedure ignores the underlying structure of the generated signal, requiring the HMCA to later expend distance calculations catching up. The other two initialisation methods do not suffer from this, and neither does k-Medoids. These results illustrate how an initialisation method acting counter to the underlying structure of the dataset can lead to significantly worse performance.

6. DISCUSSION & CONCLUSION

In this study we investigated the effect five initialisation methods from literature have on the performance of the HMCA and k-Medoids. The five initialisation methods are defined, implemented, and applied to both the HMCA and k-Medoids, the latter extended to make the comparison fair. The comparison evaluates both algorithms on both effectiveness (ability to find good clusterings) and effectiveness (ability to find them fast).

An analysis of the experimental results leads us to conclude that the HMCA has superior performance, both in effectiveness and efficiency, over k-Medoids. Particularly the *Sample* and *Cluster* initialisation methods were found to have superior performance, on all three datasets. Using these initialisation methods the HMCA was capable of consistently finding global optimal clusterings, while using substantially less computational effort to do so.

The performance of the *KA0* and *KA1* initialisation methods were particularly disappointing, for both algorithms. These initialisation methods are often used because they are deterministic, limiting stochasticity in the experimental process. However, both methods are also expensive, and become exponentially so as the number of data objects in the dataset increase. We found that this became a prohibitive factor for the 16-QAM dataset, and moreover found that the increased computational effort did not lead to a subsequent performance increase for *both* algorithms.

When computation effort available is limited (as it almost always is), additional effort spend on initialisation reduces the amount available for evolving the generated clusterings. This must therefore be offset by the quality of the generated clusterings. Our findings show this did not work for the *KA0* and *KA1*, and, to a lesser extend, for the *Uniform* initialisation methods. In contrast, the *Sample* and *Cluster* initialisation methods *did* get this balance right for the HMCA, and, given their superior performance with k-Medoids as well, we expect that this will be the case for other algorithms as well.

Limited available space has lead us to, thus far, neglect one questions: which HMCA variants have the best performance for which dataset. A succinct answer would go as follows: the curves shown in Figure 1 for the HMCA are compound curves, showing the performance of all 175 HMCA variants merged into one curve. In [14] we found that the best performing HMCA variants were those that were relaxed (more random) about cluster and data object selection, but specific (deterministic) about label selection. Analysis of the results for this study did not alter that conclusion, i.e., similar HMCA variants had the best performance in [14], as well as here.

REFERENCES

[1] L. Kaufman and P.J. Rousseeuw, *Finding Groups in Data – An Introduction to Cluster Analysis*, Wiley, Canada, 1990.

[2] B.S. Everitt, S. Landau, and M. Leese, *Cluster Analysis*, Arnold Publishers, 2001.

[3] J.P. Bigus, *Datamining with Neural Networks: solving business problems–from application development to decision support*, McGraw-Hill, 1996.

[4] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, 1988.

[5] G. Mecca, S. Raunich, and A. Pappalardo, "A new algorithm for clustering search results," *Data and Knowledge Engineering*, vol. 62, pp. 504–522, 2007.

[6] P. Baldi and S. Brunak, *Bioinformatics – The Machine Learning Approach*, MIT Press, 2nd ed. edition, 2001.

[7] P.M. Bertone-Gerstein, "Integrative data mining: The new direction in bioinformatics – machine learning for analyzing genome-wide expression profiles," *IEEE Engineering in Medicine and Biology*, vol. 20, pp. 33–40, 2001.

[8] F. Valafar, "Pattern recognition techniques in microarray data analysis: A survey," *Annals of New York Academy of Sciences*, vol. 980, pp. 41–64, 2002.

[9] Basel Abu-Jamous, Rui Fa, David J. Roberts, and Asoke K. Nandi, "Paradigm of tunable clustering using binarization of consensus partition matrices (bi-copam) for gene discovery," *PLoS ONE*, vol. 8, no. 2, 2013.

[10] L.J. Arabie, G. Hubert, and P. DeSoete, *Clustering and Classification*, World Scientific, 1999.

[11] C. Fralley and A.E. Raftery, "How many clusters? which clustering method? answer via model-based cluster analysis," *The Computer Journal*, vol. 41, pp. 578–588, 1998.

[12] E. Falkenauer, *Genetic Algorithms and Grouping Problems*, John Wiley & Sons, 1998.

[13] E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas, and A.C.P.L.F. de Carvalho, "A survey of evolutionary algorithms for clustering," *IEEE Trans. on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol. 39, no. 2, pp. 133–155, 2009.

[14] B.G.W. Craenen, A.K. Nandi, and T. Ristaniemi, "A novel heuristic memetic clustering algorithm," in *IEEE International Workshop on Machine Learning for Signal Processing*, Southampton, UK, Sept 22–25 2013.

[15] J.M Pẽna, J.A. Lozano, and P. Larrãnaga, "An emperical comparison of four initialisation methods for the k-means algorithm," *Pattern Recognition Letters*, vol. 20, no. 10, pp. 1027–1040, 1999.

[16] F. Neri, C. Cotta, and P. Moscato, Eds., *Handbook of Memetic Algorithms*, vol. 379 of *Studies in Computational Intelligence*, Springer, 2011.

[17] R.A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[18] I.W. Evett and E.J. Spiehler, *Rule Indiction in Forensic Science*, Central Research Establishment, Home Office Forensic Science Service, 1987.