Medoid-based Data Clustering with Estimation of Distribution Algorithms

Henry E. L. Cagnini,
Rodrigo C. Barros,
Christian V. Quevedo
Faculdade de Informática
Pontifícia Universidade
Católica do Rio Grande do Sul
Porto Alegre, RS, Brazil
rodrigo.barros@pucrs.br

Márcio P. Basgalupp
ICT-UNIFESP
Inst. de Ciência e Tecnologia
Universidade Federal de SP
S. J. dos Campos, SP, Brazil
basgalupp@unifesp.br

ABSTRACT

Data clustering is the machine learning task that aims at arranging data into groups (clusters) of objects according to a similarity criterion. From an optimisation perspective, it is a particular kind of NP-hard grouping problem, thus attracting much attention from the evolutionary computation community. In this paper, we propose a novel data clustering algorithm based on a univariate estimation of distribution algorithm, namely Clus-EDA. It employs a medoid-based representation in which the cluster prototypes necessarily coincide with objects from the dataset. We compare Clus-EDA with both traditional non-evolutionary clustering algorithms such as k-means and hierarchical agglomerative clustering, and also with an evolutionary algorithm for clustering, in artificial and synthetic datasets. Our results show that Clus-EDA often outperforms the baseline algorithms with regard to distinct cluster validity criteria.

CCS Concepts

•Computing methodologies → Cluster analysis; Bio-inspired approaches;

Keywords

Data Clustering, Estimation of Distribution Algorithms, Evolutionary Computation

1. INTRODUCTION

Data clustering is a task whose purpose is to determine a finite set of categories (clusters) to describe a dataset only taking into account the similarity among its objects, with no supervision whatsoever regarding the number or type of categories [9]. Examples of real-world applications that benefit from data clustering include image segmentation [2] and bioinformatics [6, 1].

From an optimisation viewpoint, data clustering is considered a particular kind of NP-hard grouping problem, thus attracting a number of studies that explore general-purpose meta-heuristics for providing an approximate solution in feasible time. Given that evolutionary algorithms (EAs) are a class of meta-heuristics widely believed to be effective on NP-hard problems, many researchers approached the data clustering problem by designing specific EAs for evolving partitions of clusters in a variety of application domains [7].

Typically, EAs for clustering employ the label-based strategy, in which an integer encoding is usually used. Each gene represents a dataset object with a value over the alphabet $\{1, 2, ..., k\}$ for a k-clustering problem, indicating the cluster each object belongs to. Approaches that employ this strategy can assume either a fixed or a variable number of clusters. Medoid-based EAs are much less frequent. In such an approach, there are strategies that encode each individual as a binary string, indicating whether or not an object is a prototype (medoid). Others encode individuals as a k-sized integer vector, where each gene represents a cluster and the integer value indicates which object is the medoid of the corresponding cluster. Regardless of the approach, medoid-based EAs have employed so far a fixed number of clusters strategy. We argue that this is not a good solution, because it assumes the user knows *a priori* the correct number of clusters, which is not what happens in real-world problems.

In this paper, we propose a novel EA for data clustering following the medoid-based approach with a variable number of clusters, namely Clus-EDA (Clustering with Estimation of Distribution Algorithms). Our approach makes use of a univariate estimation of distribution algorithm (EDA) for evolving clustering partitions following the binary string encoding. Our hypothesis is that a medoid-based EDA is capable of achieving better results than traditional data clustering algorithms such as k-means [10] and hierarchical agglomerative clustering [9]. Furthermore, we believe our

SAC 2016,April 04-08, 2016, Pisa, Italy
©2016 ACM. ISBN 978-1-4503-3739-7/16/04...$15.00
DOI: http://dx.doi.org/10.1145/2851613.2851900

approach is capable of outperforming a label-based EA called F-EAC [6], without requiring the number of clusters prior to its execution.

This paper is organised as follows. Section 2 presents our novel evolutionary approach for data clustering. Section 3 describes the methodology that we employed for the experimental analysis, as well as a discussion on our findings. We present our conclusions and future work direction in Section 4.

2. CLUS-EDA

Clustering with Estimation of Distribution Algorithms (Clus-EDA) is an EDA for medoid-based data clustering. EDAs are a particular class of evolutionary algorithms that explore the space of candidate solutions by building and sampling explicit probabilistic models of promising solutions [5]. The main characteristic of EDAs is the absence of the nature-inspired operators during evolution. Instead, the future populations are generated by learning and simulating a probability distribution from fitness-based selected individuals of the current population [13].

Clus-EDA samples solutions encoded by a univariate probabilistic model, which is responsible for determining whether each object in a dataset is a *medoid* or not. A medoid is a cluster representative, and the number of medoids indicate the number of clusters found by Clus-EDA. Each individual in Clus-EDA is a binary vector of size n, where n is the number of objects in the dataset.

Clus-EDA is a univariate EDA, also regarded as a univariate marginal distribution algorithm (UMDA) [11]. It employs a probability vector $p = (p_1, p_2, ..., p_n)$ as its probabilistic model, where p_i denotes the probability of object $\mathbf{x}_i$ to be a medoid. To learn the probability vector, each p_i is set to the proportion of 1s in the population of selected individuals.

For initialising the probability vector with some prior knowledge regarding the application domain, we execute k-means [10] multiple times varying k from 2 to $\sqrt{n}$ and select the number of clusters k^* from the partition that optimises a given clustering validity index. This heuristic is a thumb rule for defining the optimal value of k for methods that require this definition *a priori*. Even though Clus-EDA does not require setting a fixed number of clusters prior to its execution, we set $p_i = k^*/n$, $\forall i \in \{1, 2, ..., n\}$. By doing so, we potentially reduce the search-space of Clus-EDA, even though it will automatically adjust the probability vector throughout evolution, being capable of discovering partitions with any number of clusters. The clustering validity criterion used to define k^* is the Silhouette Width Criterion [14], a widely-used index to validate data clustering partitions.

In each generation of Clus-EDA, we employ the truncation method for selection, which chooses $t\%$ of the fittest individuals of that particular generation to update the probabilistic univariate model. Once the model is updated, Clus-EDA samples the probability vector p to generate an entire novel population of individuals that fully replace the previous generation. The iteration continues until a maximum number of generations is achieved.

2.1 From Individuals to Clustering Partitions

For decoding the individuals into partitions, the first step is to identify which objects are defined as medoids. Note that the number of clusters is variable since it is constantly updated according to the EDA's probabilistic model. For each non-medoid object $\mathbf{x}_i$, Clus-EDA computes the Euclidean distance between $\mathbf{x}_i$ and every single medoid, and finally assigns $\mathbf{x}_i$ to the cluster represented by its closest medoid.

The binary encoding adopted by Clus-EDA has several advantages over other typical encodings in evolutionary clustering problems. For instance, let us consider the case of the integer encoding in which each gene (object) has a value over the alphabet $\{1, 2, ..., k\}$. Such an encoding is naturally redundant (1-to-many), since there are $k!$ different genotypes that represent the same solution [7]. Furthermore, it assumes the number of clusters k is previously known, which is often not the case in real world applications.

2.2 Fitness Function

The fitness function in Clus-EDA should be capable of evaluating the quality of the data clustering partition. However, the validation of clustering structures is said to be most difficult and frustrating part of cluster analysis, to the point in which it is compared to a "black art" [8].

Several clustering validity criteria have been proposed in the specialised literature throughout the years. We refer the interested reader to a thorough survey on clustering validity criteria by Vendramin et al. [15]. Most of these criteria, however, are computationally costly.

Let n be the number of objects and a the number of attributes in the dataset. The cost of most validity criteria is quadratic in the number of objects – e.g., Dunn's ($O(an^2)$), Silhouette Width Criterion ($O(an^2)$), Gamma ($O(an^2 + n^4/k)$), McClain-Rao ($O(an^2)$), just to name a few.

Hence, we decided to choose as fitness function a validity criterion whose complexity is linear in n, namely the Simplified Silhouette Width Criterion ($SSWC$) [6]. It is an efficient implementation of the traditional Silhouette Width Criterion (SWC) [14] as follows:

$$SWC = \frac{1}{n} \sum_{i=1}^{n} \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \qquad (1)$$

where $a(i)$ is the average dissimilarity between the i^{th} object and its cluster, and $b(i)$ is the average dissimilarity between the i^{th} object and the nearest neighbor cluster. For singletons, the ratio is not computed (it is replaced by zero). The difference between the simplified and traditional Silhouette Width Criterion is in how $a(i)$ and $b(i)$ are computed. Whereas SWC computes the average dissimilarity by employing all objects belonging to the corresponding cluster (complexity of $O(an^2)$), $SSWC$ computes the average dissimilarity by using the cluster prototypes instead (complexity of $O(ank)$). Note that $SSWC$ can become costly for $k \approx n$, though we know that in practical terms $k << n$.

Table 1: Results summary. Values for the true number of clusters (k), the estimated number of clusters (k^*), Simplified Silhouette Width Criterion, Davies-Bouldin index, and Adjusted Rand Index for k-means, UPGMA, F-EAC, and Clus-EDA. Values for Clus-EDA and F-EAC are averages of 30 executions.

		k-means				UPGMA				F-EAC				Clus-EDA			
Dataset	k	k^*	SWC	DB	ARI	k^*	SWC	DB	ARI	k^*	SWC	DB	ARI	k^*	SWC	DB	ARI
s1	15	16.00	0.63	0.61	0.90	19.00	0.51	0.63	0.85	15.00	**0.71**	0.46	0.87	15.07	**0.71**	**0.42**	**0.99**
s2	15	14.00	0.61	**0.48**	0.89	15.00	0.52	0.68	0.91	15.00	**0.63**	0.57	0.87	15.07	0.62	0.53	**0.93**
s3	15	14.00	0.41	**0.69**	0.62	15.00	0.19	0.91	0.69	15.00	**0.49**	0.76	**0.86**	14.73	**0.49**	0.70	0.73
s4	15	17.00	0.47	**0.68**	0.64	18.00	0.10	0.98	0.61	15.00	**0.48**	0.77	**0.85**	15.20	0.47	0.73	0.65
sin1	6	5.00	0.63	0.53	0.67	6.00	**0.65**	0.71	**0.84**	6.00	**0.65**	0.60	0.67	6.00	**0.65**	**0.52**	**0.84**
sin2	6	6.00	0.54	1.11	0.43	5.00	**0.69**	0.62	**0.67**	5.00	**0.69**	0.48	0.55	5.00	**0.69**	**0.43**	**0.67**
sin3	6	4.00	0.45	0.89	0.44	5.00	**0.73**	0.47	**0.54**	4.00	0.72	0.44	0.44	4.00	0.72	**0.43**	0.54
sin4	6	6.00	0.51	1.06	0.64	8.00	**0.70**	0.91	0.83	6.93	0.69	0.62	0.67	6.00	0.69	**0.43**	**0.84**
sin5	6	6.00	0.55	0.78	0.65	5.00	**0.64**	0.70	0.67	8.00	**0.64**	0.57	0.67	6.00	0.63	**0.56**	**0.83**

3. EXPERIMENTAL ANALYSIS

In this section, we detail the datasets that are employed in the experiments (Section 3.1), as well as the clustering algorithms that participate in the analysis (Section 3.2), the parameters used in Clus-EDA and in the baseline algorithms (Section 3.3), and the evaluation measures to validate the results (Section 3.4). At the end of the section, we discuss the results of the empirical analysis (Section 3.5).

3.1 Datasets

For validating our results, we make use of a total of 9 datasets. The first 4 of them, namely s1, s2, s3, and s4, are artificial datasets proposed by Fränti and Virmajoki [4]. These datasets are 2-d data with $n = 5000$ and $k = 15$ Gaussian clusters with different degrees of cluster overlapping. The advantage of using artificial data is that we possess the "golden truth", i.e., the partition with the correct cluster assignments, so we can evaluate the clustering algorithms more objectively.

The second set of datasets we make use were created by Yeung et al. [16], which simulate data from microarray. The 5 synthetic microarray datasets, namely sin1, sin2, sin3, sin4, and sin5, are formed by $n = 400$ genes and $a = 20$ measurements (attributes). There are approximately 6 clusters with equal size in each of these dataset.

3.2 Baseline Algorithms

For comparison purposes in the empirical analysis, we make use of well-known clustering algorithms, namely k-means [10] and UPGMA [9], as well as F-EAC [12], which is a mutation-based EA (no crossover is performed whatsoever), with specialised mutation operators for the clustering task.

3.3 Parameters

Both k-means and UPGMA have a single parameter, which is the final number of clusters k. For deciding which value of k to use, we executed both algorithms for each dataset varying the number of clusters within $[2, \sqrt{n}]$, and we selected the value of k from the partition that optimised the SWC. This thumb rule is often used for defining the number of clusters in algorithms such as k-means.

Regarding F-EAC and Clus-EDA, we executed both within a cycle of 500 individuals and 500 generations. We kept the remaining default parameters of EAC [12]. For Clus-EDA, the only parameters are the value of the truncation selection, which we set to $t = 50\%$, and the value of the initial probability for each gene in the probabilistic vector (for generating a uniform distribution). As detailed in Section 2, we defined the initial probability as k^*/n, where k^* is the same value of k found by k-means in the thumb rule. Hence, the values of initial probability for Clus-EDA in the 9 datasets are as follows: s1 = 0.0032, s2 = 0.0028, s3 = 0.0028, s4 = 0.0034, sin1 = 0.0125, sin2 = 0.015, sin3 = 0.01, sin4 = 0.015, and sin5 = 0.015.

3.4 Evaluation Measures

Considering that all datasets that are used during the empirical analysis are synthetic, one of the evaluation measures we compute is the Adjusted Rand Index (ARI). ARI verifies the compatibility between the generated partition (henceforth called "clusters") and the reference partition (henceforth called "classes"). It is a measure *adjusted for chance*, i.e., when comparing two random partitions it yields a value close to zero. We also evaluate the results according to other two criteria, namely SWC and DB. SWC is the original Silhouette Width Criterion [14] without the prototype simplification for speeding it up, the latter being used in Clus-EDA's fitness function. Criterion DB is the Davies-Bouldin index [3], which is also an internal validity criterion that analyses the data alone.

3.5 Results

We executed Clus-EDA and F-EAC 30 times by varying the seed of each execution, since they are evolutionary non-deterministic approaches. UPGMA and k-means were executed once per number of clusters, which was varied within $[2, \sqrt{n}]$. Then, we selected the partition that optimised the SWC validity index [14] for each one of them. Table 1 presents a summary with the experimental results.

Our first analysis in this round of experiments was regarding the number of clusters found by each method. Note that Clus-EDA presents the lowest average absolute error (0.40) regarding the estimated number of clusters, followed by F-EAC and k-means, and then by UPGMA. In other words, Clus-EDA is the algorithm that best estimates the number of clusters, though we are aware that simply estimating the proper number of clusters is not enough for a clustering algorithm to be deemed *effective*.

Therefore, our second analysis was regarding the Adjusted Rand Index (ARI). Note once again that Clus-EDA seems to be the best option among the algorithms that were executed.

It provides the largest ARI value in 7 of the 9 datasets, even though it ties with UPGMA in three of them. By presenting the best ARI values, Clus-EDA demonstrates it has the greatest potential to approximate the golden truth provided by each of these datasets.

Our next analysis was regarding the internal validity criteria SWC and DB. Regarding SWC, Clus-EDA together with F-EAC and UPGMA shared wins, with a small advantage to F-EAC overall. In terms of the DB index, Clus-EDA once again has shown to be the best option, winning in 6 out of 9 datasets (with k-means winning in the remaining three datasets). Hence, we have showed that Clus-EDA is not only the best clustering algorithm in estimating the correct number of clusters, but that it also is the best algorithm regarding both external and internal clustering validity criteria.

4. CONCLUSIONS

This work proposed a novel estimation of distribution algorithm for medoid-based clustering, namely Clus-EDA. The proposed approach employs a simple but efficient and effective evolutionary framework that estimates a univariate marginal distribution model to define cluster prototypes. To guide the iterative refinement of the probabilistic model, Clus-EDA employs a clustering internal validity criterion that has complexity $O(an)$, i.e., linear in the number of objects and attributes.

We compared Clus-EDA with k-means [10] and hierarchical agglomerative clustering [9], and also with an evolutionary algorithm F-EAC [12]. For comparison purposes, we employed 9 clustering datasets: 4 of them were artificially generated based on Gaussian clusters [4], and 5 of them simulate microarray gene expression data [16]. Results show that Clus-EDA can generate data partitions that provide a greater agreement regarding the reference partitions, outperforming the baseline clustering algorithms in both external and internal clustering validity criteria. As future work, we intend to verify whether more sophisticated probabilistic models would wield improved results for medoid-based clustering.

Acknowledgment

The authors would like to thank CAPES and FAPESP for funding this research. We would also like to gently acknowledge the authors of F-EAC for providing us with its executable file.

5. REFERENCES

[1] R. C. Barros, R. Cerri, A. A. Freitas, and A. C. de Carvalho. Probabilistic clustering for hierarchical multi-label classification of protein functions. In H. Blockeel, K. Kersting, S. Nijssen, and F. Zelezny, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 8189, pages 385–400. Springer Berlin Heidelberg, 2013.

[2] K.-S. Chuang, H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen. Fuzzy c-means clustering with spatial information for image segmentation. *Computerized Medical Imaging and Graphics*, 30(1):9 – 15, 2006.

[3] D. L. Davies and D. W. Bouldin. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[4] P. Fränti and O. Virmajoki. Iterative shrinking method for clustering problems. *Pattern Recognition*, 39(5):761 – 775, 2006.

[5] M. Hauschild and M. Pelikan. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1(3):111–128, Sept. 2011.

[6] E. R. Hruschka, R. J. G. B. Campello, and L. N. de Castro. Evolving clusters in gene-expression data. *Information Sciences*, 176(13):1898–1927, 2006.

[7] 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 Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 39(2):133–155, 2009.

[8] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[9] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Willey & Sons, 1990.

[10] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. California, USA, 1967.

[11] H. Mühlenbein and G. Paaβ. From recombination of genes to the estimation of distributions i. binary parameters. volume 1141 of *Lecture Notes in Computer Science*, pages 178–187. Springer Berlin Heidelberg, 1996.

[12] M. C. Naldi, R. J. G. B. Campello, E. R. Hruschka, and A. C. P. L. F. de Carvalho. Efficiency issues of evolutionary k-means. *Applied Soft Computing Journal*, 11(2):1938–1952, Mar. 2011.

[13] J. M. Peña, J. A. Lozano, and P. Larrañaga. Unsupervised Learning Of Bayesian Networks Via Estimation Of Distribution Algorithms: An Application To Gene Expression Data Clustering. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12:63–82, 2004.

[14] P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(0):53 – 65, 1987.

[15] L. Vendramin, R. J. G. B. Campello, and E. R. Hruschka. Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, 3(4):209–235, 2010.

[16] K. Y. Yeung, M. Medvedovic, and R. E. Bumgarner. Clustering gene-expression data with repeated measurements. *Genome Biology*, 4(5):R34, Apr. 2003.