# ON THE DESIGN OF POPULATIONAL CLUSTERING

Leonardo Ramos Emmendorfer
*Centro de Ciências Computacionais, Federal University of Rio Grande, CEP 96201-900, Rio Grande, Brazil*

Keywords: Supervised clustering, Building blocks, Diversity.

Abstract: The application of clustering algorithms for partitioning the population in evolutionary computation is discussed. Specific aspects which characterize this task lead to opportunities which can be explored by the clustering algorithm. A supervised clustering algorithm is described, which illustrates the exploration of those opportunities.

## 1 INTRODUCTION

The role of the population in evolutionary computation (EC) is to maintain the information acquired during the search. The translation of the potentially huge amount of information represented by the successive populations into useful knowledge which can help guiding the search has motivated the adoption of statistical learning models, under the framework of *estimation of distribution algorithms* (Muhlenbein and Paaβ, 1996)(Etxeberria and Larrañaga, 1999), and of machine learning techniques like in (Michalski, 2000) and (Miquélez et al., 2004).

Clustering the population into partitions is one of the most widely adopted learning approaches in evolutionary computation. The application of clustering in the EC context has many aims: maintaining niches in order to preserve diversity and preventing premature convergence; improving multimodal problem solving as in (Peña et al., 2005) and (Streichert et al., 2003), detecting promising areas of the search space as in (Oliveira et al., 2004) or to improve building blocks and revealing the problem structure as in (Emmendorfer and Pozo, 2009). In all those cases, clustering was shown to be highly useful tool in EC.

An important issue about the application of clustering in EC is the computational cost, since the total number of generations of a single run might be very high. Any competent clustering method could be applied, but evolutionary computation has some features which should be better explored when choosing or designing the clustering algorithm to be used in this context. This careful exploration might allow one to design a computationally less expensive clustering algorithm without loss of effectiveness in the task.

The first aspect we point out is the sequentiality of EC, since a given population usually maintains some degree of similarity with the previous one. From a machine learning perspective, the sequence of populations in EC can be modeled as a data stream. Another aspect which should be better understood is about how accurate must the clustering algorithm be when applied to EC. Since more generations are to come, this might be an opportunity to be relaxed when trying to find the best partitioning at every generation. This hypothesis being true, computational requirements would potentially be reduced. A third aspect to be explored is that it is relatively easy to detect, even manually, the correct partitioning of small populations. The information supplied by this supervisory data provides a great opportunity for a novel class of clustering algorithms called *learning from cluster examples* or *supervised clustering* to be applied when greater populations are required.

In supervised clustering, the algorithm is trained with labeled data before applied to unlabeled data (Finley and Joachims, 2005). Labeled data corresponds to correct partitioning of complete data sets. A slight difference exists between general supervised learning and supervised clustering. In supervised clustering, only supervisory information regarding to which objects should be grouped is provided, and there is not a predefined set of classes, as required by general supervised learning (Kamishima and Motoyoshi, 2003).

This work discusses the application of supervised clustering in the population of evolutionary algorithms. In this context, a supervised clustering algorithm can be provided with full labeled data about previous populations. If, for instance, the inten-

Ramos Emmendorfer L..
ON THE DESIGN OF POPULATIONAL CLUSTERING.
DOI: 10.5220/0003114202890292
In *Proceedings of the International Conference on Evolutionary Computation* (ICEC-2010), pages 289-292
ISBN: 978-989-8425-31-7

tion is to split the population into groups possessing different building blocks as in (Emmendorfer and Pozo, 2009), then supervisory information is the presence/absence of each building block for each individual a given population. After trained, the supervised clustering algorithm would be able to find partitions for greater populations under similar conditions, for similar problems.

Those few points about opportunities related to the application of clustering algorithms in evolutionary computation lead to the description of a clustering algorithm which can be more relaxed than others, should deal with incremental data and also explore the potentially abundant supervisory information available from the execution of evolutionary algorithms on known problems. This paper proposes an algorithm which attempts to fulfill those requirements, potentially increasing the effectiveness of the clustering task when applied to EC.

## 2 SUPERVISED CLUSTERING

The search for the best partition of a given set of data points is not a straightforward task. Even when a distance is known, many possible answers about what is the correct clustering might be all equally likely. Unsupervised clustering is an ill-defined task if we do not restrict the criteria used to characterize a good clustering (Romer et al., 2004). The bias resulting from the clustering algorithm behavior can, more or less explicitly, impose some restrictions and guide the search to one of the possible answers.

Several definitions exist for what is a good partitioning. An unsupervised clustering algorithms follows an specific definition and tries to find partitions which respect criteria defined *a priori*.

In supervised clustering, on the other hand, the definition of good or bad clustering is implicit, hidden under the available labeled data. Supervised clustering is the task of automatically adapting a clustering algorithm, which learns to cluster with the aid of a training set consisting of item sets and complete partitionings of those item sets (Finley and Joachims, 2005). A clustering algorithm is trained using known "good" partitions of previously stored data. If the algorithm generalizes well, it will be able find clusters when unlabeled data is provided. This technique avoids most of the subjective aspects of clustering, since the user beliefs about expected answers are expressed in training data.

A popular technique for solving supervised clustering is based on building a binary classifier from pairwise relations observed in data (Iii et al., 2005). For a given input set, a binary classifier is trained on all pairs of input data points. The class of each pair of points is the binary information about the actual comembership of that pair. The answer of the classifier can be used as a metric or taken as the evidence that a given pair of data points should be clustered together. This learned metric is then adopted using some conventional clustering algorithm, like k-means.

Depending on how specific the attributes are, the binary classifier will not be able to generalize to other domains. Usually, the classifier is built upon the original attributes, what makes generalization restricted to data which comes from the same domain as training data comes from. Density-based derived attributes alleviate this problem, since the notion of density is not tied to a specific set of attributes.

## 3 A SUPERVISED CLUSTERING ALGORITHM APPLIED TO EVOLUTIONARY COMPUTATION

This section illustrates one possible scheme for the design of a supervised clustering algorithm which explores some of the specific aspects of evolutionary computation. The implications of the algorithm and its adoption in EC are discussed.

The algorithm is trained over some small populations which were already clustered adequately. A viable approach is to select a smaller instance of the same problem, or a similar one, then run the evolutionary algorithm in order to obtain a small population. Each individual of the population must be (manually or automatically) labeled, according to what one believes to be the best clustering. For instance, if the intention is to solve multimodal problems, then each cluster corresponds to a different optimum.

A probabilistic model is inferred from pairwise information about co-membership. Each pair of data points from the training set has a binary label which is 1 if both points belong to the same cluster and 0 otherwise. Additionally, a pairwise neighborhood must be defined, which defines the local region around any given pair of points. Many alternatives might be tested. The Gabriel Graph (Urquhart, 1982) already defines a neighborhood for a pair of points: it is related to the smallest hyperspherical region centered in the the median between the pair of points, which includes those points. Attributes such as the density of points in the neirghborhood will be computed. Other supervised clustering algorithms like in (Kamishima and Motoyoshi, 2003) also adopt attributes like this.

The relation between the density attributes $A(D_a, D_b)$ (which are defined for all pairs of points $(D_a, D_b)$) and the co-membership label for all pairs of data points in the training data set might be modeled by a logistic regression, conditioned to the satisfaction of the assumptions of the logistic model. The answer of the model is the estimated probability $P(c(D_a) = c(D_b))$ that $D_a$ and $D_b$ should be together, or $\hat{J}(c(D_a) = c(D_b))$, defined over two data points $D_a$ and $D_b$, where $c(X)$ designates the cluster label of a data point $X$.

Once the probabilistic model is defined, one can get a partitioning for unobserved data by following the supervised clustering approach.

Algorithm 1 shows a general framework which is designed for the specific application of partitioning the population during the execution of an evolutionary algorithm. It follows a simple agglomerative approach, guided by the probabilistic decision model $\hat{J}(D_a, D_b)$ for pairs of points $D_a, D_b$ in a very straightforward fashion. It puts together in the same cluster points with higher co-membership evidence $\hat{J}(D_a, D_b)$. For each labeled point $D_i$, the evidence $\hat{J}(i, .)$ which motivated the setting of that label is preserved as $E(D_i)$. A point changes its cluster label only if the new evidence is greater than the previous greatest one, stored in $E(D_i)$.

**Algorithm 1.** A simple supervised clustering approach (SSC).

*Training*: The model for $\hat{J}(D_i, D_j)$ is obtained from full example partitions of some data sets.
*Initialization*: Each data point is (i) in its own cluster initially, or (ii) cluster labels for some points are given.
Set all $E(D_i)$s to zero.
**while** convergence criteria were not met **do**
  Randomly choose a pair of points $D_a$ and $D_b$, where $c(D_a)$ is a smaller cluster than $c(D_b)$
  **if** $\hat{J}(D_a, D_b) > E(D_b)$ and $\hat{J}(D_a, D_b) > 0.5$ **then**
    $c(D_a) \leftarrow c(D_b)$
    $E(D_a) \leftarrow \hat{J}(D_a, D_b)$
  **end if**
**end while**

Additionally, two points are clustered together only if the evidence for that is greater than 0.5.

Individuals which stay in the population from one generation to another can keep their cluster labels. This preserves relevant information about clustering. An explicit bias against small clusters is adopted, in order to minimize the number of final clusters.

Convergence criteria might be related to the permanence over time of a stable distribution of cluster labels. Obviously, experimental verification will answer how fast is the convergence and how accurate is the answer.

The incremental aspect of EC is explored, since the initialization accepts some previously labeled individuals, which stay in the population due to elitism. Also, the opportunity for training the algorithm with small labeled populations is being attended by the supervised architecture.

## 4 CONCLUSIONS

This paper discusses the application of clustering in evolutionary computation, and points out opportunities which can be explored in order to design an effective clustering algorithm which is specially adapted for the application.

For illustrating this cross study, an algorithm is proposed, which explores some of the aspects pointed out. The proposed algorithm must be empirically compared to state-of-the-art clustering algorithms when applied to the population partitioning task. The effect on the performance of an evolutionary algorithm will be measured for all algorithms compared.

Although any conclusions can only be obtained after validation, there is already some evidence about the usefulness of this study. In (Emmendorfer and Pozo, 2009), a k-means clustering algorithm is continuously applied to the population of a evolutionary algorithm. Only few incremental steps of k-means are reported to be enough in order to keep centroids updated. Empirical validation must verify if the same performance is obtained in the supervised approach presented here.

## REFERENCES

Emmendorfer, L. R. and Pozo, A. T. R. (2009). Effective linkage learning using low-order statistics and clustering. *IEEE Transactions on Evolutionary Computation*, 13(6):1233–1246.

Etxeberria, R. and Larrañaga, P. (1999). Global optimization using bayesian networks. In *Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 332–339.

Finley, T. and Joachims, T. (2005). Supervised clustering with support vector machines. In *ICML '05: Proceedings of the twenty-second international conference on Machine Learning*, pages 217–224.

Iii, H. D., Marcu, D., and Cohen, W. (2005). A bayesian model for supervised clustering with the dirichlet process prior. *Journal of Machine Learning Research*, 6:1577.

Kamishima, T. and Motoyoshi, F. (2003). Learning from cluster examples. *Machine Learninig*, 53(3):199–233.

Michalski, R. S. (2000). Learnable evolution model: Evolutionary process guided by machine learning. *Machine Learning*, 38(1):9–40.

Miquélez, T., Bengoetxea, E., and Larrañaga, P. L. (2004). Evolutionary computation based on Bayesian classifiers. *International Journal of Applied Mathematics and Compututer Sciences*, 14(3):335–349.

Muhlenbein, H. and Paaβ, G. (1996). From recombination of genes to the estimation of distributions: I binary parameters. In *Parallel Problem Solving from Nature III*, pages 178–187.

Oliveira, A. C. M. and Lorena, L. A. N (2004). Detecting promising areas by evolutionary clustering search. In *Advances in Artificial Intelligence. Springer Lecture Notes in Artificial Intelligence Series*, pages 385–394.

Peña, J., Lozano, J., and Larrañaga, P. (2005). Globally multimodal problem optimization via an estimation of distribution algorithm based on unsupervised learning of bayesian networks. *Evolutionary Compututation*, 13(1):43–66.

Romer, R. R., Achan, K., and Frey, B. (2004). Learning to cluster using local neighborhood structure. In *ICML '04: Proceedings of the twenty-first international conference on Machine Learning*.

Streichert, F., Ulmer, H., and Zell, A. (2003). A clustering based niching ea for multimodal search spaces. In *Proceedings of the 6th International Conference on Artificial Evolution*.

Urquhart, R. (1982). Graph theoretical clustering based on limited neghbourhood sets. *Pattern Recognition*, 15(3):173-187.