Clustering sensory inputs using NeuroEvolution of Augmenting Topologies

David Kadish
IT University of Copenhagen
Copenhagen S, Denmark
École polytechnique fédérale de Lausanne
Lausanne, Switzerland
davk@itu.dk

ABSTRACT

Sorting data into groups and clusters is one of the fundamental tasks of artificially intelligent systems. Classical clustering algorithms rely on heuristic (k-nearest neighbours) or statistical methods (k-means, fuzzy c-means) to derive clusters and these have performed well. Neural networks have also been used in clustering data, but researchers have only recently begun to adopt the strategy of having neural networks directly determine the cluster membership of an input datum. This paper presents a novel strategy, employing NeuroEvolution of Augmenting Topologies to produce an evoltionary neural network capable of directly clustering unlabelled inputs. It establishes the use of cluster validity metrics in a fitness function to train the neural network.

CCS CONCEPTS

• **Information systems** → **Clustering**; • **Computing methodologies** → **Knowledge representation and reasoning**; *Neural networks*; • **Applied computing** → Engineering; Agriculture;

KEYWORDS

clustering, NEAT, Calinski-Harabaz, k-means

ACM Reference Format:
David Kadish. 2018. Clustering sensory inputs using NeuroEvolution of Augmenting Topologies. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3205651.3205771

1 INTRODUCTION

Grouping similar data and experiences is a fundamental building block of learning. Without the need for *a priori* information about the meaning of a particular input, clustering forms the basis for generating meaningful categorizations [3]. Given its foundational role in learning, it is perhaps surprising that few efforts have used neural networks to perform clustering and none could be found that use neuroevolution. This paper develops a technique called NEAT Clustering (NEAT-CLU) for clustering using neuroevolution.

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205771

2 BACKGROUND

Clustering refers to the process of assembling unlabeled data into like groups. A number of clustering strategies are well-established: k-nearest neighbours (k-NN), k-means, fuzzy c-means (FCM), DBSCAN, and self organizing maps (SOM) are standard tools for cluster analysis [2]. Of these, only SOMs use neural networks (NNs), generating a neural map that is overlaid on the input data. These methods share an underlying model of input data plotted on a hyperplane and the use of a distance measurement to assign a cluster.

Two existing efforts use NNs to bypass the use of a distance metric in cluster assignments and assign cluster membership as a direct output of the NN [5, 6]. They both require that data be presented in pairs with pre-determined measurements of similarity and neither employ evolutionary methods to form their networks.

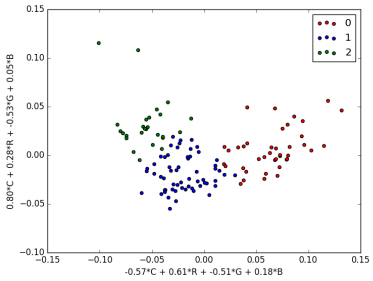
The evolutionary NN employed in this paper is known as neuroevolution of augmenting topologies (NEAT) [7]. NEAT evolves the structure and weights of its neurons in tandem, adding layers and complexity as necessary to achieve an optimal fitness. NEAT and its derivatives have been applied to a wide range of tasks but this is its first application to cluster assignment.

3 ALGORITHM

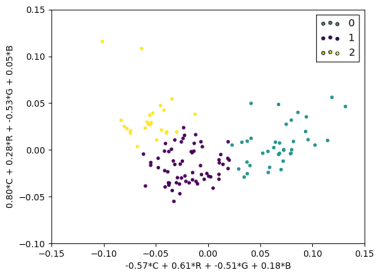
The NEAT algorithm itself is well-documented [7] and is used in its standard form in NEAT-CLU, so it is not covered in detail here. NEAT-CLU uses the raw input data, so there is no need to preprocess samples. For k clusters, the output is encoded using $\lfloor \log_2(k) \rfloor + 1$ output neurons with an unsigned step activation function. The full NN output is treated as a binary number representing the assigned cluster. When the number of binary combinations does not match the desired number of clusters, some clusters are assigned multiple binary numbers. For example, in this trial, the outputs 01 and 10 both map to the second cluster.

One of the key insights in NEAT-CLU is that clustering metrics can be a key component of the fitness function for training an evolutionary NN. The Calinski-Harabaz (CH) score is a measure of the comparison between intra-cluster variability and inter-cluster variability [1]. For a sample set of N observations, divided into k clusters with the centroid of cluster i at m_i, the CH score can be written as $\mathcal{F}_{ch} = \frac{S_B}{S_W} \times \frac{N-k}{k-1}$ where the intercluster variance S_B and intracluster variance S_W are $S_B = \sum_{i=1}^{k} n_i ||m_i - m||^2$ and $S_W = \sum_{i=1}^{k} \sum_{x \in c_i} ||x - m_i||^2$.

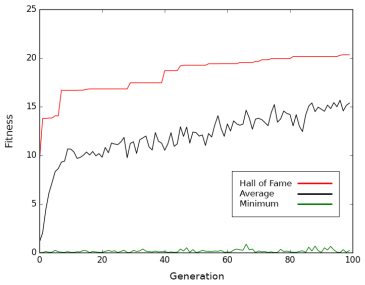
Two metrics are added to the fitness function to encourage even clustering. A demerit ($\mathcal{F}_k$) is assessed if fewer clusters are created than desired. Another ($\mathcal{F}_n$) penalizes disparity in group size. The



(a) Clusters created by the NEAT clusterer. Axes represent the two components of a PCA decomposition of the 4-dimensional input space.



(b) Clusters created by a K-means clusterer. Axes represent the two components of a PCA decomposition of the 4-dimensional input space.



(c) Fitness over generations in the evolutionary process that produced the clustering network.

Figure 1: Side-by-side comparison of NEAT-CLU and k-means clustering algorithms (a-b). Generational fitness (c).

Table 1: Cluster metric scores

	Calinski-Harabaz	Silhouette
NEAT-CLU	103.31	0.36
k-means	108.23	0.38

resulting fitness function can be written as $\mathcal{F} = w_{ch}\mathcal{F}_{ch} - w_k\mathcal{F}_k - w_n\mathcal{F}_n$ where w denotes a weighting constant, the demerit for deviation from the desired number of clusters (K) is $\mathcal{F}_k = K - k$ and a measure of cluster size disparity is $\mathcal{F}_n = k - \sum_{i=1}^{k} \frac{n_i}{\max(n)}$.

4 METHOD

This work formed part of an inquiry about how a robot could experience an ecosystem, so the experiment focused on the clustering of plant leaves. The robot was equipped with a single-pixel camera with which it R-G-B and white reflectance from different leaves. The NEAT-CLU clustering algorithm was trained on these samples and then used to sort new samples into three distinct groups. This result was compared to a k-means clustering of the same data.

5 RESULTS

Using the NEAT-CLU algorithm, the robot sorted the leaves into three clusters. The NEAT network was evolved over 100 generations with a population of 100 individuals (figure 1c). The results of NEAT-CLU are shown (figure 1a) beside the results of k-means clustering (figure 1b). NEAT-CLU and k-means produce similar results with minor differences at the cluster boundaries. NEAT-CLU has effectively learned to closely emulate a k-means clustering strategy. The cluster scores for the two methods are quite close (Table 1) though k-means fares slightly better in both evaluated metrics.

6 DISCUSSION

The slight underperformance of NEAT-CLU and its increased complexity with respect to k-means suggest that NEAT-CLU will not replace the standard clustering tools. However, NEAT-CLU can offer a degree of flexibility that is unavailable to k-means. The structure of clusters in k-means — as well other standard clustering methods — stems from the algorithm's clustering mechanism. K-means performs well on gaussian-distributed datasets with equal-sized clusters, but often performs poorly on data that is distributed in other ways [4]. NEAT-CLU's clusters are structured by the fitness function, which can be easily modified to suit new datasets.

The CH portion of the fitness function could be replaced by another clustering metric, such as the S_Dbw cluster validity index [4]. This would allow NEAT-CLU to adapt readily to many differently structured datasets. This flexibility is the true advantage of performing clustering using neuroevolution. The same algorithm can be adjusted — even during the evolutionary process — to fit a wide variety of different datasets and end-goals.

ACKNOWLEDGMENTS

Thanks to Stefania Santagati, Per Nagbøl, Astrid Petitjean, Sebastian Risi, Thomas Bolander, Laura Beloff and Kasper Støy for their help with this work, and the anonymous reviewers for their feedback.

REFERENCES

[1] T. Calinski and J. Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods* 3, 1 (1974), 1–27. https://doi.org/10.1080/03610927408827101

[2] Adil Fahad, Najlaa Alshatri, Zahir Tari, Abdullah Alamri, Ibrahim Khalil, Albert Y. Zomaya, Sebti Foufou, and Abdelaziz Bouras. 2014. A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis. *IEEE Transactions on Emerging Topics in Computing* 2, 3 (sep 2014), 267–279. https://doi.org/10.1109/TETC.2014.2330519

[3] Martijn Goudbeek, Roel Smits, Anne Cutler, and Daniel Swingley. 2005. Acquiring Auditory and Phonetic Categories. In *Handbook of Categorization in Cognitive Science*. Elsevier, 497–513. https://doi.org/10.1016/B978-008044612-7/50077-9

[4] M. Halkidi and M. Vazirgiannis. 2011. Clustering validity assessment: finding the optimal partitioning of a data set. In *Proceedings 2001 IEEE International Conference on Data Mining*. IEEE Comput. Soc, 187–194. https://doi.org/10.1109/ICDM.2001.989517

[5] Yen-Chang Hsu and Zsolt Kira. 2016. Neural network-based clustering using pairwise constraints. In *4th International Conference on Learning Representations*. San Juan, Puerto Rico, 1–12. arXiv:1511.06321 http://arxiv.org/abs/1511.06321

[6] Federico Raue, Sebastian Palacio, Andreas Dengel, and Marcus Liwicki. 2017. Classless Association Using Neural Networks. In *Artificial Neural Networks and Machine Learning – ICANN 2017: 26th International Conference on Artificial Neural Networks, Alghero, Italy, September 11-14, 2017, Proceedings, Part II*. Springer, Cham, 165–173. https://doi.org/10.1007/978-3-319-68612-7_19

[7] Kenneth O. Stanley and Risto Miikkulainen. 2002. Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation* 10, 2 (jun 2002), 99–127. https://doi.org/10.1162/106365602320169811