Parallel Glowworm Swarm Optimization Classification Algorithm Implemented with Apache Spark

Bren Hutchinson [1] Simone Ludwig [1] Gongyi Xia [1] Benjamin Herrmann [2]

[1]North Dakota State University [2]University of Chile

Abstract

With the ever-growing relevance of big data due to the growth of the internet and technology, there is large demand for large-scale data management and classification. In this paper, we proposed an implementation of the glowworm swarm optimization classification algorithm (SCGSO) that is parallelized using the Apache Spark framework. The main idea of SCGSO is to use the capabilities of the standardized GSO algorithm in finding multi-modal solutions and apply it to to several target centroid labels, and assigning any unlabeled data points to the nearest centroid. For the experimentation, four datasets were used to evaluate the SCGSO algorithm with varying dimensionality and number of data points. The experimental results show that the algorithm performs better for lower dimensionality data sets, and scales nicely with size of dataset.

Introduction

Classification is a data mining technique in which a model is used with the intention of categorizing the data points into categorical, mutually exclusive group. As the volume of data continues to expand, so does the need for effective processing and organization of this data. Along with the increase in raw dataset size, there is also a rise in data dimensions and complexity. To address these issues, Apache Spark has emerged as a powerful framework for performing computations in a large-scale cluster of nodes. Spark offers a versatile platform for data processing and analysis, and even supports multiple programming languages such as Scala, Python, and R.

Over the past two decades, significant advancements have been made in Swarm Intelligence algorithms, which leverage nature-inspired approaches to solve optimization problems. These swarm-like algorithms draw inspiration from various natural phenomena, such as bird flocks, bee colonies, and even galaxies. A distinguishing characteristic of these algorithms lies in their lack of central member of the swarm, rather the swarm members are all equal participants in the end result.

Glowworm Swarm Optimization

Glowworm Swarm Optimization (GSO) is a relatively recent swarm intelligence model introduced by Krish-nan and Ghose in 2005. In GSO, a swarm of n glowworms are initially deployed in a random dispersion in a predetermined solution space. Each glowworm is a solution of an objective function and has a luciferin level, denoted L_j associated with it. The luciferin level is akin to the fitness of the glowworm's location. More luciferin, or a higher fitness level, represents a brighter glow for the individual and indicates a better solution. Under probabilistic mechanisms, each glowworm is only attracted to its neighbors with a more intense brightness than its own, within a local range. Then, depending on the density of its local-decision domain, it will either increase its local-decision domain to find more neighbors or it will reduce the range to split the neighborhood into numerous smaller groups. This process is then repeated until the algorithm reaches its termination condition, in which a majority of individuals gather around the brighter glowworms. In all, the GSO algorithm follows five phases: the luciferin-update phase, neighborhood-select phase, moving probability-computer phase, movement phase, and *decision radius update phase.*

In this specific paper, steps were taken to alter the algorithm into a classification problem.

Methodology

The proposed algorithm has its origins within a GSO algorithm with the intention of finding the optimal centroid vector for each target class in the given dataset. In SCGSO, each particle is permanently encoded with a worm's position and velocity as n-dimensional vectors.

Initially a swarm of glowworms of a user-set size is created. Each worm is given an identification number such that it can be referenced by other worms. Then, each glowworm is given a random position vector ($\vec{p_j}$) and velocity vector ($\vec{v_j}$) using a uniform probability distribution. Class labels are evenly distributed across worm. Then, each fitness level is calculated $F(j)$ which is used to find luciferin levels.

$$F_1(j^c) = \sum_{i=1}^{C^0} d(\vec{p_j^c}, \vec{p_i^{c^0}}) - \sum_{k=1}^{C} d(\vec{p_j^c}, \vec{p_k^c}) \tag{1}$$

Here, glowworm j^c has classification label c. C represents the number of data instances of class C, while C^0 is the number of data instances not in class C. The goal of the fitness function was to maximize the inter-centroid distances while minimizing the distances between the centroid and its respective data points.

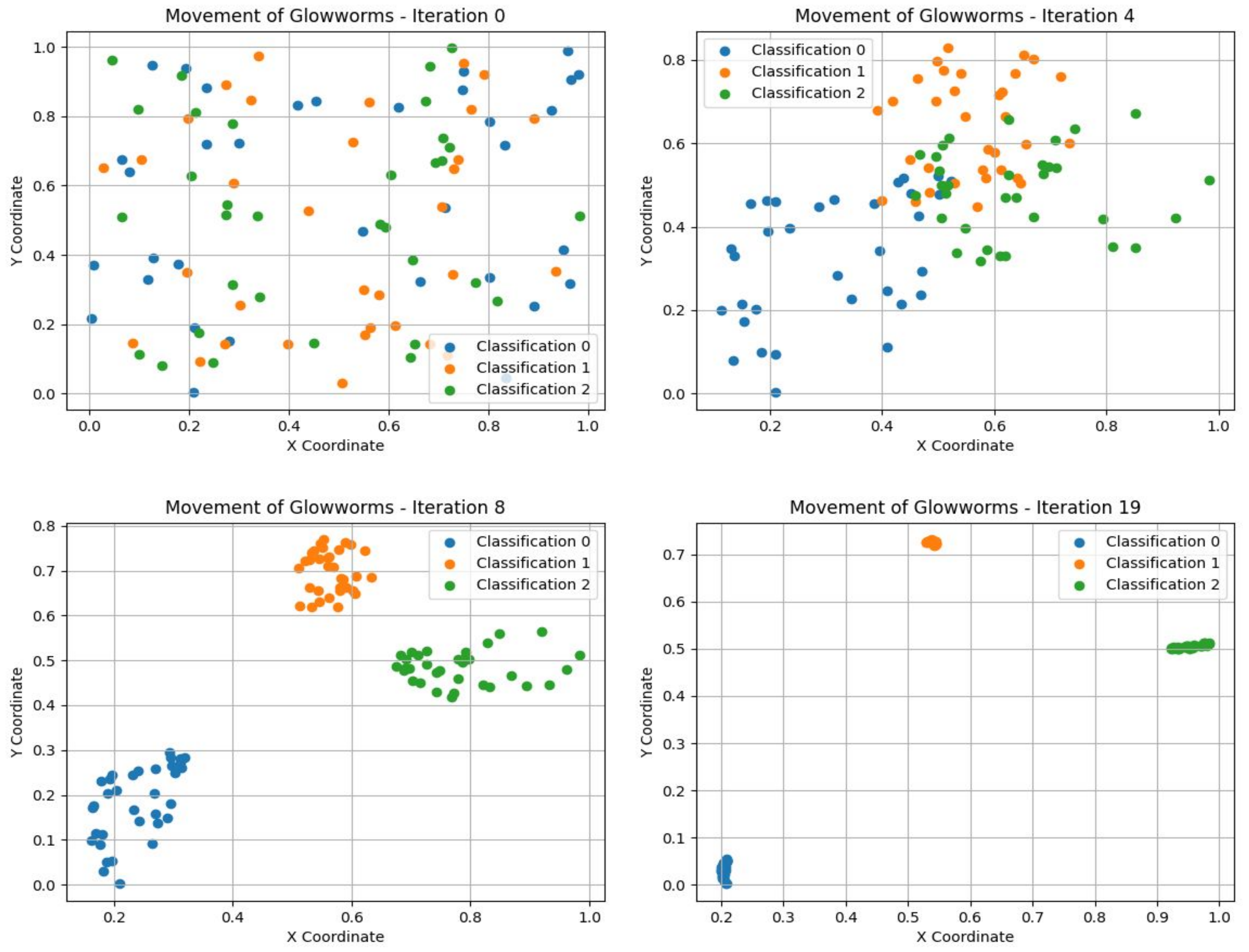

Figure 1. Figure 1-4: Glowworms throughout the converging process on an artifical dataset

Afterwards an iterative process using RDD operations on the broadcasted swarm is performed. Each iteration will update the glowworm swarm, which will be used as the input for the next iteration. Before transformations, the swarm is sent to each respective task using broadcast variables. Then, GSO constants are retrieved.
To evaluate movement direction and distance, two mapper transformations were utilized. The first transformation finds all optimal neighbors for each individual glowworm and then, using a roulette selection technique, selects a direction. The distance is the euclidean distance between the given and neighbor worm, multiplied by a step size. The second transformation applied the position update for each worm according to its velocity vector, and then updates the luciferin value accordingly.

Results

To assess the robustness of the proposed SCGSO algorithm, it was subjected to testing on multiple datasets, as listed in Table 1. The evaluation involved measuring two key metrics: accuracy and average time per iteration, both presented in Table 2.

Table 1. Statistics on Datasets

Dataset	Instances	Features	Training Instances	Testing Instances	Class Labels
Iris	150	4	120	30	3
Heart	270	13	216	54	2
Magic	19020	11	15214	3806	2
Skin	245057	4	196046	49011	2

Table 2. Performance Under Standard Features

Dataset	Accuracy (%)	Average Time Per Iteration (s)
Iris	94.59	0.1042
Heart	78.52	0.1427
Magic	76.19	0.1083
Skin	89.70	1.13

Acknowledgements

The authors acknowledge the support of the NDSU Department of Computer Science sponsored by NSF Directorate for Engineering and EPSCoR Co-Funding through NSF grant EEC-2050175. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] https://www.databricks.com/spark/about.

[2] Nailah Al-Madi, Ibrahim Aljarah, and Simone A Ludwig. Parallel glowworm swarm optimization clustering algorithm based on mapreduce. In *2014 IEEE symposium on swarm intelligence*, pages 1–8. IEEE, 2014.

[3] Jamil Al-Sawwa and Mohammad Almseidin. A spark-based artificial bee colony algorithm for unbalanced large data classification. *Information*, 2022.

[4] Jamil Al-Sawwa and Simone A Ludwig. Parallel particle swarm optimization classification algorithm variant implemented with apache spark. *Concurrency and Computation: Practice and Experience*, 32(2):e5451, 2020.

[5] Ibrahim Aljarah and Simone A Ludwig. Parallel particle swarm optimization clustering algorithm based on mapreduce methodology. In *2012 fourth world congress on nature and biologically inspired computing (NaBIC)*, pages 104–111. IEEE, 2012.

[6] Ivanoe De Falco, Antonio Della Cioppa, and Ernesto Tarantino. Facing classification problems with particle swarm optimization. *Applied Soft Computing*, 7(3):652–658, 2007.

[7] Zhengxin Huang and Yongquan Zhou. Using glowworm swarm optimization algorithm for clustering analysis. *Journal of Convergence Information Technology*, 6(2):78–85, 2011.

[8] Wissem Inoubli, Sabeur Aridhi, Haithem Mezni, Mondher Maddouri, and Engelbert Mephu Nguifo. An experimental survey on big data frameworks. *Future Generation Computer Systems*, 86:546–564, 2018.

[9] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, 39:459–471, 2007.

[10] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

[11] KN Krishnanand and Debasish Ghose. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm intelligence*, 3:87–124, 2009.

[12] Goutham Miryala and Simone A Ludwig. Comparing spark with mapreduce: glowworm swarm optimization applied to multimodal functions. *International Journal of Swarm Intelligence Research (IJSIR)*, 9(3):1–22, 2018.

[13] Salman Salloum, Ruslan Dautov, Xiaojun Chen, Patrick Xiaogang Peng, and Joshua Zhexue Huang. Big data analytics on apache spark. *International Journal of Data Science and Analytics*, 1:145–164, 2016.

https://www.ndsu.edu/cs/ brenmhutchinson@gmail.com