An Efficient Social Spider Optimization for Data Clustering using Data Vector Representation

T. Ravichandran, B. Janet, A. V. Reddy

Abstract: ***In this article, we propose a new clustering algorithm namely an efficient social spider optimization for data clustering using data vector representation (ESSODCDI). It uses a data vector representation for each spider so that its memory requirements can be reduced.***

Unlike other nature-inspired algorithms, it requires lesser memory requirements. We find that its clustering results are by far better than those of other nature-inspired algorithms.

Keywords: Nature-inspired algorithms, Social Spider Optimization, Clustering, Memory Requirements, Data Vector.

I. INTRODUCTION

The mathematical model for data clustering can be described as follows.

Let F be the data clustering function from a D dimensional dataset DS having R data instances to a collection of K clusters C such that

$$DS = \{dv_1, dv_2, \dots, dv_i, .., dv_R\}$$

where data instance dv_i is a set of attributes such that

$$dv_i = \{attribute_1, attribute_2, \dots, attribute_i, \dots, attribute_D\}$$

and

$$C = \{(c_1, cent_1), (c_2, cent_2), \dots \dots, (c_i, cent_i), \dots, (c_K, centK)\}$$

where c_i is i^{th} cluster of data instances and $cent_i$ is centroid of i^{th} cluster. The clusters in C satisfy the following conditions.

$c_i \neq \varnothing$, $\bigcap_{i=1}^{K} c_K = \varnothing$,

$\bigcup_{i=1}^{K} c_K =$ DS, and $1 \leq K \leq R$.

Unique feature of Meta heuristic algorithms is different methods of search process. Meta heuristic optimization algorithms are used to solve wide range of real-time problems due to the following reasons.

- their simplicity
- they do not need slope information
- they avoid local optima
- they can be exploited in an ample range of problems wrapping different disciplines.

SSO was applied for solving data clustering problem. Later its memory requirements were reduced in SSODCSC. In this article, a new algorithm that using data vector representation for each spider is proposed. Section II describes Social Spider Optimization, Section III describes the proposed algorithm. Experimental results are specified in Section IV. Conclusion is specified in Section V.

II. SOCIAL SPIDER OPTIMIZATION

In a social spider colony, each spider, depending on its gender, performs a various tasks such as designing communal web, mating, killing the other spiders etc.

The communal web acts as both communicational channel and common environment. The spiders use vibrations to pass information in the communal web. (Cuevas et. al, 2013) by taking inspiration from social spiders proposed SSO. In SSO, each spider is considered as a possible solution in n-dimensional space. A spider becomes the globally best spider s_{gbs} if the weights of all other spiders are less than its weight. Likewise, a spider becomes the worst spider s_{ws} if all spiders are having more weight than it. The weight of a spider scan be computed using equation (1).

$$w[s] = \frac{fitness(s) - fitness(s_{ws})}{fitness(s_{gbs}) - fitness(s_{ws})} \quad (1)$$

A. Constructing solution space

The dimensions of all spiders are initialized using equation (2). lower and upper functions return the littlest and the greatest in the domain of of the dim^{th} attribute.

$$spid[s, dim] = lower(dim) + random(0, 1) * (upper(dim) - lower(dim)) \quad (2)$$

B. Evaluating subsequent positions of spiders

The next positions of the spiders mainly depends on the weights and distances of spiders with highest fitness values, spiders at nearest distance with better fitness, and nearest female spiders. The amount of vibrations that spider s_j produces to spider $s_i$can be estimated using equation (3).

$$vibrations[s_i, s_j] = w[s_j] * e^{-dist(s_i, s_j)^2} \quad (3)$$

C. Evaluating subsequent locations of female spiders

The subsequent location of a spider s_f is based on weight and distance of spider having best fitness value and spiders having better fitness at nearest distance. The subsequent location of a female spider s_f that attracts the other spider is calculated as per equation (4). If it runs away from the other spiders, its next position can be found as per equation (5).

$$spid[s_f, dim] = spid[s_f, dim] + r_1 * (spid[s_f, dim] - spidsgbs, dim * wsgbs * e - distsf, sgbs2 + r2 * spidsf, dim - spidsnbs, dim * wsnbs * e - distsf, snbs2 + r3 * r4 - 0.5 \quad (4)$$

$$spid[s_f, dim] = spid[s_f, dim] - r_1 * (spid[s_f, dim] - spidsgbs, dim * wsgbs * e - distsf, sgbs2 - r2 * spidsf, dim - spidsnbs, dim * wsnbs * e - distsf, snbs2 + r3 * r4 - 0.5 \quad (5)$$

Revised Manuscript Received on March 10, 2020.

T. Ravichandran, Department of Computer Applications, National Institute of Technology, Trichy, India.

B. Janet, Department of Computer Applications, National Institute of Technology, Trichy, India.

A. V. Reddy, Department of Computer Applications, National Institute of Technology, Trichy, India.

D. Evaluating subsequent locations of male spiders

The subsequent location of a s_{dm} is computed as per equation (6).

$$spid[s_{dm}, dim] = spid[s_{dm}, dim] + r_1 * (spid[s_{dm}, dim] - spidsnfs, dim * wsnfs * e - distsdm, snfs2 + r3 * r4 - 0.5 \quad (6)$$

The vibrations from a female spider s_{nfs} at minimum distance plays important role in estimating the subsequent position of male spiders that have better fitness values. The weighted mean of spiders whose gender is male, W is used to compute subsequent positions of male spiders having low fitness values. It is obtained as per equation (7). Then the female spiders can be represented as $s_{f_1}, s_{f_2}, s_{f_3}, \ldots\ldots, s_{f_{N_f}}$ and the male spiders can be represented as $s_{m_1}, s_{m_2}, s_{m_3}, \ldots\ldots s_{m_{N_m}}$.

$$W = \frac{\sum_{j=1}^{Nm} spid[s_{m_j} dim] * w[s_{m_j}]}{\sum_{j=1}^{Nm} weight[s_{m_j}]} \quad (7)$$

A non-dominant male spider s_{ndm} moves to its next position as per equation (8).

$$spid[s_{ndm}, dim] = spider[s_{ndm}, dim] + r_1 * W \quad (8)$$

E. Evaluation of new spiders

Each dominant male spider makes spiders with gender female happy by offering gifts (small insects) and mates with them to give birth to new spider. In SSO, the new spider is generated using the Roulette wheel method. To maintain best population, the worst spider will be replaced by new spider, if weight of new spider is greater than that of the worst spider. (Cuevas et al., 2013) applied SSO on19 benchmark functions. They got good results for the proposed method. SSO enables the agents spread over the solution space evenly to produce global optima. More over, it enables the agents to take larger steps and smaller steps at different rates.

III. AN EFFICIENT SOCIAL SPIDER OPTIMIZATION FOR DATA CLUSTERING USING DATA VECTOR REPRESENTATION (ESSODCDI)

In SSO clustering, each spider contains K centroids. Each centroid may have 0 or more data instances close to it. Therefore, the required memory is high and execution time is more. To avoid these two issues, a single centroid representation for each spider is proposed in SSODCSC. The presentation of a spider in Fig 1A, Fig 1B and Fig 1C specify the K-centroid, single centroid and data vector representations of a spider. Single centroid representation is K times better than K-centroid representation with respect to required memory. Data vector representation is K times better than Single centroid representation with respect to required memory. Each spider represents a data instance in the data file. So, each spider will have D attribute/dimension values. Initially, each spider is initialized with a random data instance taken from the dataset. The pairs of spider ID, and data instance ID are stored in a table. K centroids are randomly initialized.

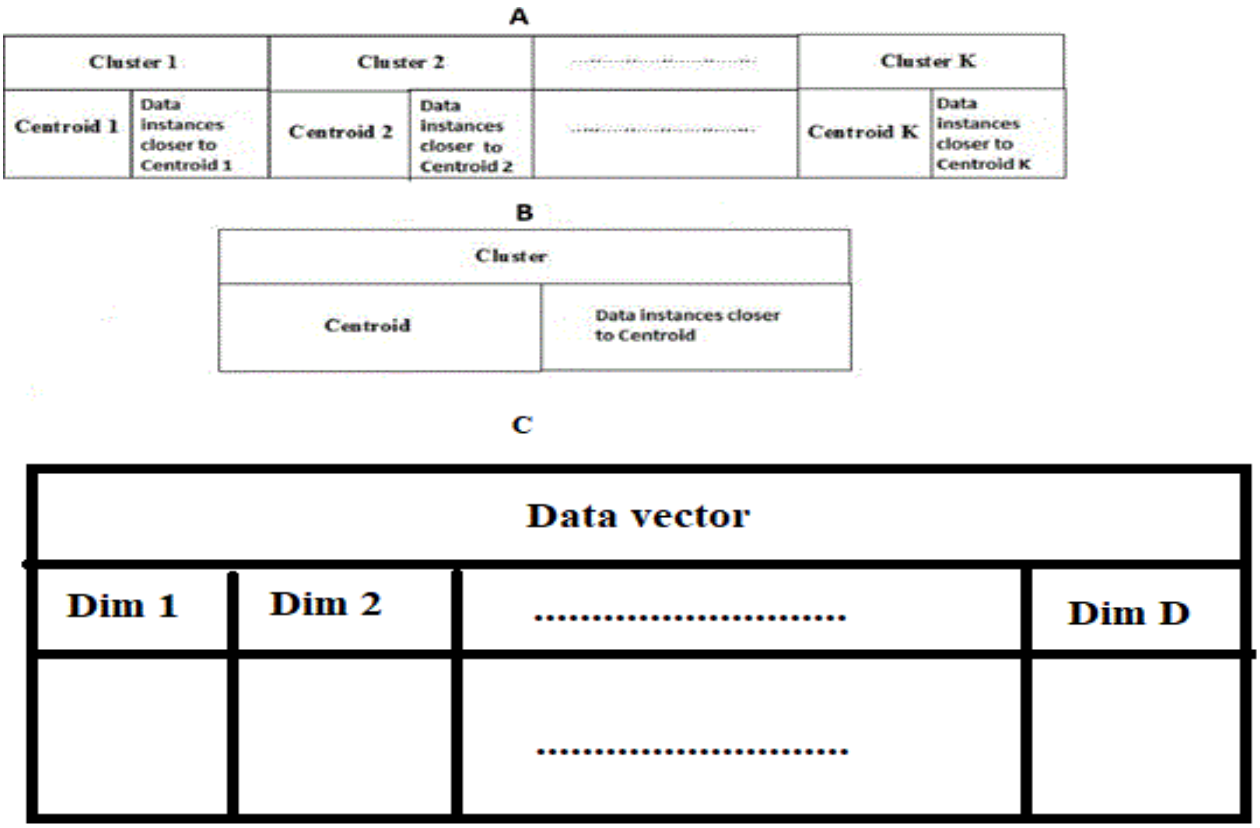

Figure 1: A. Spider with K-centroids B. Spider with one centroid C. Spider with data vector

The distances of each spider from these K centroids are found and the spider is assigned to the centroid which is at minimum distance from the spider. Then the spiders are moved to their next positions in the search space. And then dominant males are allowed to participate in the mating operation. K-centroids are updated based on the list of spiders associated with them. This process is repeated until the number of iterations exceeds the maximum limit. The fitness of a spider s is the minimum of its distances from K-centroids. Algorithm 1 specifies the steps involved in ESSODCDI.

Algorithm 1. **An efficient Social Spider Optimization for data clustering using data vector representation (ESSODCDI)**

Input: dataset

Output: K-clusters of relevant data instances

1. Read number of spiders N, threshold probability PF, and upper bound for number of iterations Max.
2. Derive number of spiders with gender female N_f, and number of spiders with gender male N_m using the following formulae.

N_f= floor[(0.90-rand(0,1)*0.25)] * N

$N_m = N - N_f$

3. For each spider s in the population

{

Initialize spider s with a randomly taken data instance dv from the dataset DS.

}

}

4. Repeat

For each spider s in the population

{

Compute its distances from all centroids and assign it to its nearest centroid.

Take the minimum of its distances from K-centroids as its fitness.

Derive the weight of spider s using its fitness.

Find its next position based on its gender.

}

For each dominant male spider s in the population

{

Find the set of spiders whose gender is female in the range of mating of dominant male spider s.

Allow dominant male spider s to mate with those female spiders to generate a new spider.

Remove spider having least weight and place new spider in its location, if new spider is better than it.

}

Update the K-centroids based on the list of associated data instances.

Increase Iteration by 1.

until number of iterations exceeds Max.

5. Return the spider having largest weight

.

IV. EXPERIMENTAL RESULTS

ESSODCDI is applied on UCI data sets. Table 1 specifies how SICD is inversely proportional to the number of iterations. For iris dataset, the SICD values are 114.04, 112.00, 108.14, 102.15 and 92.01 respectively, when number of iterations is changed from 100 to 300 in steps of 50. However, it remains at 92.0122, after 300 iterations and it becomes obvious that ESSODCDI converges in 300 iterations. The population size remains the same in all iterations of ESSODCDI, like SSO based data clustering. Table 2 consists of F-measure values produced by the clustering algorithms. The ESSODCDI produced atleast 10% better values than each algorithm. Table 3 specifies the silhouette coefficient values produced clustering algorithms when applied on UCI datasets. ESSODCDI produces best silhouette coefficient values for all datasets.

Table 1: The relationship between SICD values and the number of iterations : ESSODCDI

Dataset	100 iterations	150 iterations	200 iterations	250 iterations	300 iterations
Iris	114.04	112.00	108.14	102.15	**92.01**
Vowel	146823.11	146001.03	145366.88	145023.22	**144279.24**
CMC	6046.44	5834.11	5622.64	5289.85	**5000.59**
Glass	328.58	317.80	266.02	228.33	**194.08**
Wine	16638.99	16210.89	16013.44	15794.26	**15309.00**

Table 2: F-measure values: ESSODCDI and other clustering algorithms

Dataset	PSO	GA	ABC	IBCO	ACO	SMSSO	BFGSA	SOS	SSO	SSODCSC	ESSODCDI
Wine	78.79	70.25	72.48	63.34	64.88	60.1	67.88	63.78	78.42	94.98	**96.55**
Cancer	83.42	71.38	70.55	62.98	60.34	61.95	62.03	64.8	74.34	96.49	**98.19**
CMC	51.49	55.15	57.79	51.92	50.49	51.98	52.92	52	51.45	61.01	**72.88**
Vowel	68.11	60.69	64.74	62.12	68.13	54	68.68	65.56	70.85	90.46	**94.23**
Iris	90.95	62.41	62.58	60.43	71.95	64.43	62.47	62.43	85.81	96.95	**98.66**
Glass	44.94	45.01	43.72	54.66	43.36	55.48	42.21	44.46	58.54	70.92	**75.44**

Table 3: Comparison between clustering algorithms with respect to the average silhouette coefficient values

Dataset	K-means	PSO	GA	ABC	IBCO	ACO	SMSSO	BFGSA	SOS	SSO	SSODCSC	ESSODCDI
Wine	0.6490	0.562	0.500	0.522	0.415	0.448	0.600	0.610	0.641	0.6885	0.7505	**0.7912**
Cancer	0.5894	0.622	0.527	0.584	0.449	0.485	0.599	0.565	0.599	0.6107	0.6966	**0.7688**
CMC	0.3733	0.328	0.316	0.372	0.344	0.498	0.480	0.490	0.478	0.5111	0.7889	**0.8522**
Vowel	0.4588	0.4079	0.427	0.401	0.621	0.410	0.582	0.625	0.602	0.6492	0.7148	**0.8455**
Iris	0.7099	0.716	0.438	0.473	0.604	0.505	0.625	0.579	0.651	0.6333	0.8833	**0.9423**
Glass	0.3661	0.280	0.299	0.207	0.546	0.290	0.489	0.415	0.401	0.4419	0.6264	**0.7422**

V. CONLUSION

We proposed a new clustering algorithm using social spiders. As the existing clustering algorithms based on social spider optimization require more memory space, a better representation for spiders is proposed. The proposed algorithm presented promising results, however, there are a few issues that could be addressed in future works. There is no specific mechanism to specify the gender of the spiders. The first N_f spiders are taken as female spiders and the remaining are considered as male spiders.

REFERENCES

1. **Aloise, Deshpande, Hansen, Popat**, NP-hardness of Euclidean sum-of-squares clustering, Machine Learning, volume 75, issue 2, pages 245-248, year 2009
2. **Cuevas, Cienfuegos, Zaldvar, Perez Cisneros**, A swarm optimization algorithm inspired in the behavior of the social-spider, Expert Systems with Applications, volume 40, pages 6374-6384, year 2013

3. **Steinley**, K-means clustering: A half-century synthesis, British Journal of Mathematical and Statistical Psychology, volume 59, pages 1-34, year 2006
4. **Bernabe Loranca B, Gonzalez-Velazquez R, Olivares-Benitez E, Ruiz-Vanoye J, Martinez-Flores J**, Extensions to K-Medoids with Balance Restrictions over the cardinality of the partitions, Journal of Applied Research and Technology, Volume 12, Issue 3, Pages 396-408, Year 2014
5. **Holland JH**, Genetic algorithms, Sci. Am, Volume 267, Pages 66-72, Year 1992
6. **Seyedali Mirjalili, Andrew Lewis**, Whale Optimization Algorithm, Journal of Advances in Engineering Software, Volume 25, Pages 51-67, Year 2016
7. **Kennedy J, Eberhart R**, Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks, Volume 6, Pages 1942-1948, Year 1995
8. **Dorigo M, Gambardella LM**, A cooperative learning approach to the travelling salesman problem, IEEE Transactions on Evolutionary Computation, Volume 1, Issue 1, Pages 53-66, Year 1997
9. **Fred Glover**, Future paths for integer programming and links to artificial intelligence, Computers and Operations Research, Volume 13, Issue 5, Pages 533-549, Year 1986
10. **He S, Wu QH, Saunders JR**, Group search optimizer: an optimization algorithm inspired by animal searching behavior, IEEE Transactions on Evolutionary Computation, Volume 13, Issue 5, Pages 973-990, Year 2009