J. Comp. & Math. Sci. Vol.2 (3), 415-422 (2011)

Implementation, Analysis and Performance Evaluation of Hybrid Rough Set Particle Swarm Optimization (HRSPSO) Algorithm for Image Pixel Classification and Segmentation

H. S. BEHERA

Department of Computer Science and Engineering,
Veer Surendra Sai University of Technology (VSSUT), Burla, Sambalpur,
Orissa, India.

ABSTRACT

In this paper, a comprehensive study has been made on Hybrid Rough Set Particle Swarm (HRSPSO) Algorithm for Image Pixel Classification as proposed by Das *et. al.*[4]. The HRSPSO optimization algorithm has been implemented with a novel method in MATLAB platform considering 50 iterations and 20 particles. The experimental study and performance evaluation show that HRSPS optimization Algorithm is observed to be having optimal solution with smallest DB (Davies-Bouldin) index and it converges after fifteenth iterations.

Keywords: Image Processing, Clustering, Image Segmentation, Classification, Davies-Bouldin index, Rough Set, Particle Swarm Optimization, Hybrid Rough Set Theory, Image Pixel Classification, Fuzzy C- Means (FCM).

1. INTRODUCTION

Image segmentation refers to the process of partitioning a digital image to multiple segments or set of pixels. The goal of segmentation is to simplify the representation of an image into different segments that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries in images. It is also the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. The result of image segmentation is a set of segments that collectively cover the entire image. Image segmentation is nothing but the process of dividing an image into disjoint homogenous regions[7]. These regions usually contain similar objects of interest. The homogeneity of the segmented regions can be measured using some image property such as pixel intensity. Some of the practical applications of image segmentation are Medical Imaging, Locate objects in satellite images (roads, forests, etc.), Face recognition, Fingerprint recognition, Traffic control systems, and Machine vision. A Several general-purpose algorithms and techniques have been developed for image classification[9]. Since there is no general solution to the image segmentation problem, these techniques often have to be combined with domain knowledge in order to effectively solve an image segmentation problem for a problem domain.

2. PRELIMINARIES

Clustering can be defined as the optimal partitioning of a given set of n data points into c subgroups, such that data points belonging to the same group are as similar to each other as possible[6]. The data points from two different groups share the maximum difference. Image segmentation can be treated as a clustering problem where the features describing each pixel correspond to a pattern, and each image region corresponds to a cluster. Many clustering algorithms have widely been used to solve the segmentation problem like K-means [6-10], FCM and ISODATA . Some hard clustering approaches do not consider overlapping of classes that occur in many practical image segmentation problems. For example, in remote sensing satellite images, a pixel corresponds to an area of land space may not necessarily belong to a single type of land cover, Which in turn indicates that the pixels in a satellite image can be associated with a large amount of imprecision and uncertainty. Therefore, application of the principles of fuzzy set theory has remained a popular choice for the researchers in this domain.

2.1 Rough Set Theory

The rough set theory, pioneered by Pawlak[1] in mid 1980's, has emerged as a promising mathematical tool for extracting knowledge from datasets which contain imperfection, such as noise, unknown values or errors due to inaccurate measuring equipment. Rough set theory constitutes a sound basis for discovering patterns in hidden data and thus has extensive applications in data mining in distributed systems. Recently it has been emerged as a major mathematical tool for managing uncertainty that arises from granularity in the domain of discourse—that is, from the indiscernibility between objects in a set. The intention is to approximate a rough (imprecise) concept in the domain of discourse by a pair of exact concepts, called the lower and upper approximations. These exact concepts are determined by an indiscernibility relation on the domain, which, in turn, may be induced by a given set of attributes ascribed to the objects of the domain. The lower approximation is the set of objects definitely belonging to the vague concept, whereas the upper approximation is the set of objects possibly belonging to the same.

2.2 Hybrid Rough Set Particle Swarm Optimization

Particle swarm optimization (PSO) is a swarm intelligence based algorithm to find a solution to an optimization problem in a search space, or model and predict social behaviour in the presence of objectives. It is a population based stochastic optimization technique developed by Eberhart *et. al*.[3-5] in 1995, inspired by social behaviour of bird flocking or fish schooling.PSO is similar to a genetic algorithm (GA) in that the system is initialized with a population of random solutions.Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied. It is unlike a GA, however, in that each potential solution is also assigned a randomized velocity, and the potential solutions, called *particles,* are then "flown" through the problem space.

Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. This value is called p_{best}. Another "best" value that is tracked by the *global* version of the particle swarm optimizer is the overall best value, and its location, obtained so far by any particle in the population. This location is called g_{best}. The particle swarm optimization concept consists of, at each time step, changing the velocity each particle toward its p_{best} and g_{best} locations. Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward p_{best} and g_{best} locations. There is also a *local* version of PSO in which, in addition to p_{best}, each particle keeps track of the best solution, called/best, attained within a local topological neighborhood of particles. The Hybrid Rough Set Particle Swarm Optimization (HRSPSO) technique has been used for grouping the pixels of an image in its intensity space. Medical and remote sensing satellite images become corrupted with noise very often. Fast and efficient segmentation of such noisy images has remained a challenging problem for many years. In this work, image segmentation is treated as a clustering problem. Each cluster is modeled with a rough set. PSO is employed to tune the threshold and relative importance of upper and lower approximations of the rough sets. Davies–Bouldin[2] clustering validity index is used as the fitness function, which is minimized while arriving at an optimal partitioning.

PSO, which gained huge popularity as a naturally inspired optimization tool in recent times, is used. to tune the threshold, and relative importance of upper and lower approximation parameters of the sets. The Davies–Bouldin[2] clustering validity index is used as the fitness function of the PSO, that is minimized.

The PSO algorithm, as first described by Eberhart and Kennedy[3] is reminiscent of the behavior of flock of birds or the sociological behavior of a group of people. In PSO, a population of articles is initialized with random positions:

$$\vec{Z}_i(t) = \left[\vec{Z}_{i,1}(t), \vec{Z}_{i,2}(t), \dots \vec{Z}_{i,j}(t)\right]$$

and velocities:

$$\vec{v}_i(t) = \left[\vec{v}_{i,1}(t), \vec{v}_{i,2}(t), \dots \vec{v}_{i,d}(t)\right]$$

in d-dimensional space. A fitness function, f is evaluated, using the particle's positional coordinates as input values. Positions and velocities are adjusted, and the function is evaluated with the new coordinates at each time-step. The velocity and position update equations for the p^{th} dimension of the i^{th} particle in the swarm may be given as follows:

$$v_{ip}(t+1) = \omega . v_{ip}(t) + C_1 . \varphi_1 . (P_{lip} - Z_{ip}(t)) + C_2 . \varphi_2 . (P_{gp} - Z_{ip}(t))$$
$$Z_{ip}(t+1) = Z_{ip}(t) + v_{ip}(t+1) \quad (1)$$

The variables Φ_1 and Φ_2 are random positive numbers, drawn from a uniform distribution, and with an upper limit Φ_{max}, which is a parameter of the system. C_1 and C_2 are called acceleration constants, and ω is the inertia weight. P_{li} is the best solution found so far by an individual particle, while Pg represents the fittest particle found so far in the entire community.

2.3 Rough C-Means Algorithm

In rough c-means (RCM) algorithm, the concept of c-means clustering is extended by viewing each cluster as an interval or rough set. A rough set Y is characterized by its lower and upper approximations RL (Y) and RU (Y) respectively. This permits overlaps between clusters. Here an object Xi can be part of at most one lower approximation. If $X_k \in$ R1(Y) of cluster Y , then simultaneously $X_k \in$ R2(Y) .If X_i is not a part of any lower approximation, then it belongs to two or more upper approximations. Here the cluster center Z_i of cluster $_{Ci}$ is computed as:

$$\vec{Z_i} = w_{low}\frac{\sum_{\vec{X}_k \epsilon R_1(Y)} \vec{X}_k}{|R_1(Y)|} + w_{up}\frac{\sum_{\vec{X}_k \epsilon [R_2(Y)-[R_1(Y)]} \vec{X}_k}{|R_2(Y)-R_1(Y)|}$$

If $R_2(A) - R_1(A) \neq \emptyset$

$$= w_{low}\frac{\sum_{\vec{X}_k \epsilon R_1(Y)} \vec{X}_k}{|R_1(Y)|} \quad (2)$$

where the parameters W_{low} and W_{up} correspond to the relative importance of the lower and upper approximations respectively. Here |R1(Y)| indicates the number of pattern points in the lower approximation of cluster Y, while |R2(Y)-R1(Y)|is the number of elements in the rough boundary lying between the two approximations. In RCM (Rough c-means), a threshold parameter needs special mention. If the difference of distances (Euclidean usually) of an object X_k from two cluster centers Zi and Z_j of clusters Ci and Cj respectively, is lesser than some threshold δ, then $X_k \in R_2(C_j)$ and $X_k \in R_2(C_i)$ and X_k cannot be a member of any lower approximation. Else, $X_k \in R_1(C_j)$ such that distance $d(X_k,Z_i)$ is minimum over the c clusters.

Choice of Parameters on RCM

It is observed that the performance of the algorithm is dependent on the choice of W_{low}, Wup and threshold δ. W_{low} = 1− Wup , 0.5 < W_{low} < 1 and 0 < δ < 0.5 is allowed. It is to be noted that the parameter threshold measures the relative distance of an object X_k from a pair of clusters having centroids Zi and Z_j. The smaller the value of threshold, the more likely is X_k to lie within the rough boundary (between upper and lower approximations) of a cluster. This implies that only those points which definitely belong to a cluster (lie close to the centroid) occur within the lower approximation. A large value of threshold implies a relaxation of this criterion, such that more patterns are allowed to belong to any of the lower approximations. The parameter W_{low} controls the importance of the objects lying within the lower approximation of a cluster in determining its centroid. A lower W_{low} implies a higher W_{up}, and hence an increased importance of patterns located in the rough boundary of a cluster towards the positioning of its centroid.

2.4 Tuning the Cluster Validity Index with PSO

In this work we employed a PSO algorithm to determine the optimal values of the parameters W_{low} and δ for each c (number of clusters). For the fitness function of the PSO, we have chosen a statistical mathematical function, also called a cluster validity index, well known as Davies-Bouldin (DB) index . This measure is a function of the ratio of the sum of within-cluster scatter to between-cluster separation, and it uses both the clusters and their sample means. First, we define the within i-th cluster scatter and the between i-th and j-th cluster distance respectively as,

$$S_{i,q} = \left[\frac{1}{N}\sum_{\vec{X}\epsilon C_i}\left\|\vec{X}-\vec{Z}_i\right\|_2^q\right]^{1/q}$$

$$d_{ij,t} = \left\{\sum_{p=1}^{d}|Z_{i,p}-Z_{j,p}|^t\right\}^{1/t} = \left\|\vec{Z}_i-\vec{Z}_j\right\|_t$$

where q, t >= 1, q is an integer and q, t can be selected independently. Ni is the number of elements in the i-th cluster Ci. Next Ri,qt is defined as,

$$R_{i,qt} = max_{j\epsilon K, j\neq i}\left\{\frac{S_{i,q}+S_{j,q}}{d_{ij,t}}\right\}$$

Finally, we define the DB measure as,

$$DB(c) = \frac{1}{c}\sum_{i=1}^{c}R_{i,qt}$$

The smallest DB(c) indicates a valid optimal partition.

3. HRSPS ALGORITHM

We treat all the pixels of an input image as data points. The grey scale intensity of each pixel serve as a single feature. Although the data points are single dimensional, the number of data items is as high as 65, 636 for a 250X250 grey image. Then we run a Rough C-Means (RCM) algorithm on the image pixel data. Parameters of the RCM are evolved by employing a PSO algorithm. We find that this results into an excellent image segmentation algorithm having two advantages. It removes noisy spots and it is less sensitive to noise than other techniques.

3.1 Steps of the HRSPS Optimization Algorithm:

1. Choose the initial mean Z_i for the clusters.
2. Initialize a population (array) of particles with random positions and velocities on ***d*** dimensions in the problem space.
3. For each particle, evaluate the desired optimization fitness function in d variables.
4. Compare particle's fitness evaluation with particle's pbest. If current value is better than pbest, then set pbest value equal to the current value, and the pbest location equal to the current location in d-dimensional space.
5. Compare fitness evaluation with the population's overall previous best. If current value is better than gbest, then reset gbest to the current particle's array index and value.
6. Change the velocity and position of the particle according to equations **(1)** and *(2),* respectively:

   a. *Yid = Vid + CI* * rand0 * *(Pid-xld)*

b. + *cz* * Rand0 * *L& -xid)* (1)

c. xld= Xid + Vid (2)

7. Loop to step (2) until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations (generations).

3.2 Pseudo Code

For each particle
Initialize particle
END
Do
For each particle
Calculate fitness value
If the fitness value is better than the best fitness value (p_{Best}) in history
set current value as the new p_{Best}
End
Choose the particle with the best fitness value of all the particles as the g_{Best}
For each particle
Calculate particle velocity according equation (a)
Update particle position according equation (b)
End

4. EXPERIMENTAL STUDY

4.1 Implementation

The Hybrid Rough Set - PSO algorithm for image pixel classification is implemented. First of all, a noisy image is taken as input and converted into a gray scale image. The image pixels are classified using the K-means clustering algorithm, which gives the initial means and their respective positions in the clusters for the required number of clusters. After obtaining the initial values for the cluster centers, the upper and the lower bounds of each cluster are calculated and the respective cluster centers are upgraded using the Rough C-means algorithm. The threshold parameter (δ) and the other parameters W_{low} and W_{up} are tuned using the Particle Swarm Optimization technique. A statistical mathematical function, called the Davies-Bouldin (DB) index is used as the fitness function of the PSO. Finally, the smallest DB index is obtained which is observed to be the optimal solution. The convergence of the solution is plotted against the number of iterations.

4.2 Results

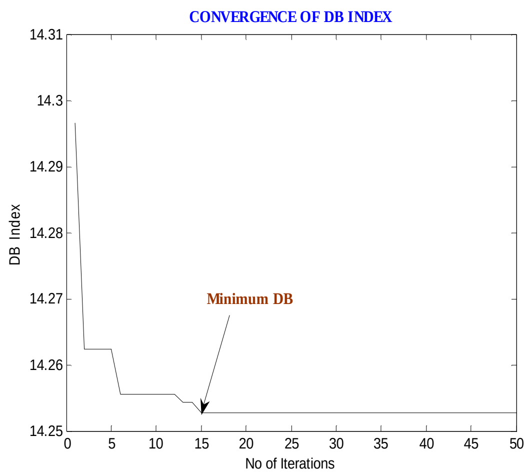
No of Iterations = 50. No of Particles = 20.

No. of Iterations	Threshold value(δ)	Parameter w_{low}	Minimum DB Index
1	1.0000e-003	0.6889	$14.2967\text{x}10^{-3}$
2	0.0013	0.6389	$14.2623\text{x}10^{-3}$
3	0.0050	0.6000	$14.2623\text{x}10^{-3}$
4	0.0033	0.6000	$14.2623\ \text{x}10^{-3}$
5	0.0036	0.6000	$14.2623\ \text{x}10^{-3}$
6	0.0022	0.6000	$14.2556\ \text{x}10^{-3}$
7	0.0029	0.6000	$14.2556\ \text{x}10^{-3}$
8	0.0037	0.6000	$14.2556\ \text{x}10^{-3}$
9	0.0028	0.6000	$14.2556\ \text{x}10^{-3}$
10	0.0035	0.6000	$14.2556\ \text{x}10^{-3}$
11	0.0040	0.6000	$14.2556\ \text{x}10^{-3}$
12	0.0037	0.6000	$14.2556\ \text{x}10^{-3}$
13	0.0028	0.6000	$14.2544\ \text{x}10^{-3}$
14	0.0028	0.6000	$14.2544\ \text{x}10^{-3}$
15	0.0038	0.6000	$14.2527\ \text{x}10^{-3}$
16	0.0038	0.6000	$14.2527\ \text{x}10^{-3}$
17	0.0031	0.6000	$14.2527\ \text{x}10^{-3}$
18	0.0030	0.6000	$14.2527\ \text{x}10^{-3}$
19	0.0030	0.6000	$14.2527\ \text{x}10^{-3}$
20	0.0042	0.6000	$14.2527\ \text{x}10^{-3}$
21	0.0028	0.6000	$14.2527\ \text{x}10^{-3}$
22	0.0037	0.6000	$14.2527\ \text{x}10^{-3}$
23	0.0036	0.6000	$14.2527\ \text{x}10^{-3}$
24	0.0040	0.6000	$14.2527\ \text{x}10^{-3}$
25	0.0029	0.6000	$14.2527\ \text{x}10^{-3}$
26	0.0042	0.6000	$14.2527\ \text{x}10^{-3}$
27	0.0030	0.6000	$14.2527\ \text{x}10^{-3}$
28	0.0036	0.6000	$14.2527\ \text{x}10^{-3}$
29	0.0039	0.6000	$14.2527\ \text{x}10^{-3}$
30	0.0037	0.6000	$14.2527\ \text{x}10^{-3}$
31	0.0035	0.6000	$14.2527\ \text{x}10^{-3}$
32	0.0035	0.6000	$14.2527\ \text{x}10^{-3}$
33	0.0036	0.6000	$14.2527\ \text{x}10^{-3}$
34	0.0035	0.6000	$14.2527\ \text{x}10^{-3}$
35	0.0036	0.6000	$14.2527\ \text{x}10^{-3}$
36	0.0036	0.6000	$14.2527\ \text{x}10^{-3}$
37	0.0036	0.6000	$14.2527\ \text{x}10^{-3}$

38	0.0035	0.6000	14.2527 x10^{-3}
39	0.0035	0.6000	14.2527 x10^{-3}
40	0.0035	0.6000	14.2527 x10^{-3}
41	0.0035	0.6000	14.2527 x10^{-3}
42	0.0035	0.6000	14.2527 x10^{-3}
43	0.0035	0.6000	14.2527 x10^{-3}
44	0.0036	0.6000	14.2527 x10^{-3}
45	0.0035	0.6000	14.2527 x10^{-3}
46	0.0035	0.6000	14.2527x10^{-3}
47	0.0035	0.6000	14.2527 x10^{-3}
48	0.0036	0.6000	14.2527 x10^{-3}
49	0.0036	0.6000	14.2527 x10^{-3}
50	0.0036	0.6000	14.2527 x10^{-3}

MINIMUM DB INDEX

= 0.0142527≈0.0143



CONCLUSION

A Comprehensive study has been made on HRSPSO algorithm after implementation with considering 50 iteration and 20 particles. The experimental results and performance evaluation show that HRSPS optimization Algorithm is observed to be having good performance.

REFERENCES

1. Pawlak, Z., Rough Sets, Theoretical Aspects of Reasoning about Data. Kluwer Academic, Dordrecht, (1991).
2. Davies, D.L., Bouldin, D.W., A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence 1, 224–227, (1979).
3. Eberhart, R. C., Shi, Y.: Particle swarm optimization: Developments, applications and resources, In Proceedings of IEEE International Conference on Evolutionary Computation, Vol. 1,81-86, (2001).
4. Swagatam Das[1], Ajith Abraham[2] and Subir Kumar Sarkar[1] “A Hybrid Rough Particle Swarm Algorithm for Image Pixel Clasification”
5. Kennedy, J, Eberhart R. Particle swarm optimization, In Proceedings of IEEE International Conference on Neural Networks, 1942-1948 (1995).
6. Jain, A.K., Murty M.N., and Flynn P.J. Data Clustering: A Review, *ACM Computing Surveys*, *Vol* 31, No. 3, 264-323 (1999).
7. Trivedi M. M, Bezdek J. C, Low-level segmentation of aerial images with fuzzy clustering, *IEEE Trans. on Systems, Man and Cybernetics, Volume* 16, Issue 4 July, (1986).
8. L.O. Hall, I.B. Özyurt, J.C. Bezdek, Clustering with a genetically optimized approach, *IEEE Trans. Evo Comp*. 3 (2) 103–112 (1999).
9. Bandyopadhyay, S., and Maulik, U. Genetic clustering for automatic evolution of clusters and application to image classification, Pattern Recognition, 35, 1197-1208 (2002).

10. Lingras, P., West, C., Interval set clustering of Web users with rough k-means. Technical Report No. 2002-002, Department of Mathematics and Computer Science, St. Mary's University, Halifax, Canada, (2002).