Combining IWC and PSO to Enhance Data Clustering

Ahmed Z. Skaik, Wesam M. Ashour

Computer Engineering Department, The Islamic University of Gaza

Gaza Strip, Palestine, 2017

ahmskaik@gmail.com , washour@iugaza.edu.ps

Abstract—In this paper we propose a clustering method based on combination of the Particle Swarm Optimization (PSO) and the inverse weighted clustering algorithm IWC, It is shown how PSO can be used to find the centroids of a user specified number of clusters and basically uses PSO to refine the clusters formed by IWC. Since PSO algorithm was showed to successfully converge during the initial stages of a global search, but around global optimum, the search process will become very slow. On the contrary, IWC algorithm can achieve faster convergence to optimum solution, Experimental results show that the proposed technique has much potential to improve the clustering process.

Index Terms— data clustering, particle swarm optimization, inverse weighted K-Means.

I INTRODUCTION

Data clustering is the process of grouping together similar multi-dimensional data attributes into a number of clusters or groups. Clustering algorithms have been applied to a wide range of problems, such as exploratory data analysis, data mining, pattern recognition and machine learning [1]. More specifically, objects are represented by a set of features which characterize them. The object features are usually represented as a data point in a multi-dimensional space. So clustering can be considered as partitioning of data points based on a homogeneity criterion. When the number of clusters, K, is known as a priori knowledge, clustering is formulated in such a way that objects in the same cluster being more similar in some sense than those in different clusters. The IWC algorithm, starting with k arbitrary cluster centres in space, partitions the set of given objects into k subsets based on a distance metric. The centres of clusters are iteratively updated based on optimization of an objective function. This method has been shown to be less sensitive to poor initialisation than the traditional K-Means algorithm [2]. Recently, many clustering algorithms based on evolutionary computing such as Genetic Algorithms have been introduced, and only a couple of applications used Particle Swarm Optimization [3]. Unlike the Genetic algorithm (GA), PSO does not have complicated evolutionary operators such as crossover and mutation [4]. In the PSO algorithm, the potential solutions called particles, are obtained by ‘‘flowing’’ through the problem space by following the current optimum particles. Generally speaking, the PSO algorithm has a strong ability to find the most optimistic result, but it suffers from converging to a local optimum. By suitably modulating the PSO parameters, convergence can be accelerated and the ability to find the global optimistic result can be enhanced.

idea is the fact that PSO at the beginning stage of algorithm is able to search whole space for the optimum solution and reduce the search area. When the PSO algorithm reaches to a solution roughly close to the optimum solution, the clustering process switches to IWC algorithm to finish the process faster and more accurately. A proper stage for switching the clustering process is sensed by inspecting the PSO fitness function along the process.

The paper has been organized as follows:

In the next section we show the related works in that field and in section 3 introduce IWC algorithm. In Section 4 we review standard PSO algorithm. We explain the proposed algorithm in Section 5. In Section 5.1 we present the result of experiments on synthetic and real data sets. Finally we draw the paper to the conclusion in Section 6.

II RELATED WORK

Various researches have been carried out to improve the efficiency of K-Means algorithm with Particle Swarm Optimization. Particle Swarm Optimization gives the optimal initial seed and using the best seed K-Means algorithm produces better clusters and produces much accurate results than traditional K-Means algorithm.

W. Barbakh and C. Fyfe. [5,6] proposed an enhanced methods for assigning data points to the suitable clusters and solve the problem of sensitivity to initial conditions. Shafiq Alam [7] proposed a novel algorithm for clustering called Evolutionary Particle Swarm Optimization (EPSO)-clustering algorithm which is based on PSO. The proposed algorithm is based on the evolution of swarm generations where the particles are initially uniformly distributed in the input data space and after a specified number of iterations; a new generation of the swarm evolves. Lekshmy P Chandran et al. [8] describes a recently developed Meta heuristic optimization algorithm named harmony search helps to find out near global optimal solutions by searching the entire

solution space. K-Means performs a localized searching. Chunqin Gu, Qian Tao [9] proposed a new combination between Chaotic particle swarm and K-Means which features better search efficiency than K-Means, PSO and CPSO.

III INVERSE WEIGHTED CLUSTERING

One of the most important components of a clustering algorithm is the measure of similarity used to determine how close two patterns are to one another. The IWC algorithm [10] - which solve the problem of sensitivity to initial conditions in the K-Means algorithm - groups the set of data points in space into a predefined number of clusters. In this regard, the Euclidean distance is commonly used as a similarity measure. The strategy in this algorithm is to group data points in such a way that the Euclidean distance between data points belonging to each group being minimized. The data points in each group (cluster) are represented by the group centre of mass, referred to as the cluster centroid. Hence the IWC algorithm attempts to find the best points in space as the cluster centroids.

The IWC algorithm has the following logic:

$$J_I = \sum_{i=1}^{N}\sum_{k=1}^{K} \frac{1}{\| \mathbf{x}_i - \mathbf{m}_k \|^P} \qquad (1)$$

$$\mathbf{m}_k = \frac{\sum_{i=1}^{N} b_{ik}\mathbf{x}_i}{\sum_{i=1}^{N} b_{ik}} \qquad (2)$$

Where

$$b_{ik} = \frac{1}{\| \mathbf{x}_i - \mathbf{m}_k \|^{P+2}} \qquad (3)$$

The partial derivative of J_I with respect to $\mathbf{m}_k$ will maximize the performance function J_I. Therefore, the implementation of (2) will always move $\mathbf{m}_k$ to the closest data point to maximize J_I to ∞,

However, the implementation of (2) will not identify any clusters as the prototypes [11] always move to the closest data point. But the advantage of this performance function is that it doesn't leave any prototype far from data: all the prototypes join the data.

The authors enhance this algorithm to be able to identify the clusters without losing its property of pushing the prototypes inside data by changing b_{ik} in (3) to the following:

$$b_{ik} = \frac{\| \mathbf{x}_i - \mathbf{m}_{k*} \|^{P+2}}{\| \mathbf{x}_i - \mathbf{m}_k \|^{P+2}} \qquad (4)$$

where $\mathbf{m}_{k*}$ is the closest prototype to $\mathbf{x}_i$.

With this change, they have an interesting behavior: (4) works to maximize J_I by moving the prototypes to the freed data points (or clusters) instead of the closest data point (or local cluster).

Note that (3) and (4) never leaves any prototype far from the data even if they are initialized outwith the data. The prototypes always are pushed to join the closest data points using (3) or to join the free data points using (4). But (3) doesn't identify clusters while (4) does.

(4) keeps the property of (3) of pushing the prototypes to join data, and provides the ability of identifying clusters.

The clustering process terminates when one of the following conditions is satisfied:

1. The number of iterations exceeds a predefined maximum.
2. When change in the cluster centroids is negligible.
3. When there is no cluster membership change.

IV PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is an optimization algorithm which simulates the movement and flocking of birds [12]. Particles are the agents that represent individual solutions while the swarm is the collection of particles which represent the solution space. The particles then start moving through the solution space by maintaining a velocity value $\mathbb{V}$ and keeping track of its best previous position achieved so far. This position value is known as its personal best position and denoted by vector $P_i=\{p_{i1}, p_{i2},\ldots, p_{in}\}$, and at each iteration, the velocity of particle and its new position is defined according to the following equations

$$V_i^{(t)} = w * V_i^{(t-1)} + c_1 * r_1(P_i - X_i^{(t-1)}) + c_2 * r_2(G - X_i^{(t-1)}) \qquad (5)$$

$$X_i^{(t)} = X_i^{(t-1)} + V_i^{(t)} \qquad (6)$$

Where, ω is called the inertia weight that controls the impact of previous velocity of particle on its current one. In the references [13,14], several selection strategies of inertial weight ω have been given. Generally, at the beginning stages of PSO algorithm, the inertial weight ω should decrease rapidly, once the swarm converge around the optimum solution, the inertial weight must decrease slowly. r_1 and r_2 are two independently uniformly distributed random variables in range [0,1]. c_1 and c_2 are positive constant parameters called acceleration coefficients which control the maximum step size between successive iterations.

Global best denoted by vector $G=\{g_1, g_2,\ldots, g_n\}$is another best solution which is the best fitness value which is achieved by any of the particles. The fitness of each particle or the entire swarm is evaluated by a fitness criterion. The flow chart of basic PSO is shown in Figure 1

According to Equation (5) the velocity of the particle at each iteration is calculated using three terms: the velocity of the particle at previous iteration, the distance of particle from its the best previous position and the distance from the best position of the entire population. Having the velocity of particle, the particle flies to a new position according to Equation (6). This process is repeated until a termination

condition is reached. Two common conditions used for terminating the PSO algorithm are exceeding the number of iterations from a predefined level and negligible change for particles in successive iterations.

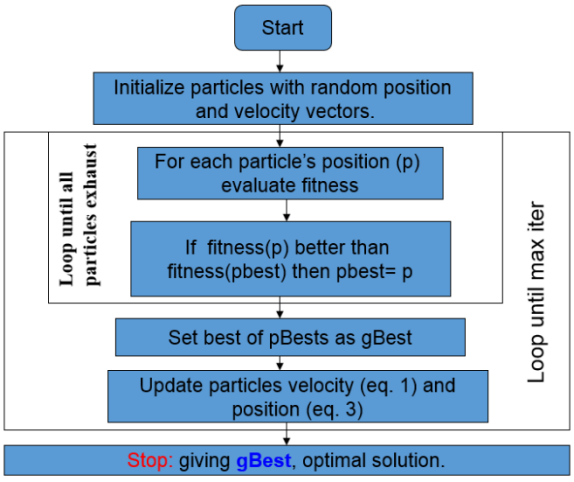

Figure 1. Basic flow diagram of PSO.

V. HYBRID PSO-IWC FOR CLUSTERING

The proposed algorithm works in two phases. Phase I is describe the Particle Swarm Optimization and how can it find the global optimal, while Phase II is describe the IWC Algorithm. The Phase I gives better seed selection and reduce the search area, since the PSO algorithm is a global search algorithm, which has a strong ability to find global optimistic result. However, the convergence speed of PSO algorithm near to the solution is very slow. The IWC algorithm, on the contrary, converge fast to a local optimum result, but its ability to find the global solution takes too many iterations. The output of Phase I is given as input to Phase II which generates the final clusters. The cluster generated by this proposed algorithm is much accurate, faster and of good quality in comparison to IWC algorithm. By combining the PSO and the IWC algorithms, a novel clustering approach is formulated in this paper. We refer to it as PSO–IWC hybrid algorithm. The motivation for combining these clustering methods is

1. Solve different distributions of centroids for multiple runs.
2. Accelerate searching for centroids by reducing searching area.
3. Dealing with multi-dimensional data (three dimensions are proposed and tested experimentally.

We start the data clustering by PSO algorithm it allows to search all space for a global solution. When the region of global optimum is found by PSO we continue the clustering using IWC. This strategy accelerates the convergence speed as well as accuracy. In this way the IWC algorithm finalizes the clustering task.

We detect the proper stage for switching from PSO to IWC, using PSO fitness function. When the value of fitness function for a number of successive iterations changes negligibly the clustering algorithm switches to IWC.

Like the PSO-KM [15] algorithm We start with initializing a group of random particles in solution space. First, all the particles are updated according to the Equations (5) and (6), until a new generation set of particles are generated. The flying particles are used to search the global best position in the solution space [16]. Finally the IWC algorithm is used to search around the global optimum. In this way, the proposed hybrid algorithm would find the optimum solution more quickly.

The procedure for this PSO–KM algorithm can be summarized as follows:

Step 1: Initialize the position and velocity of particles randomly. Each particle is a potential solution for clustering problem in hand. In the context of clustering, a single particle represents the centroid of clusters. Hence the i-th particle is initialized as follows:

$$X^{(0)}_i = (Z^{(0)}_{i1}, Z^{(0)}_{i2}, \ldots Z^{(0)}_{ik}) \quad (7)$$

Where $Z^{(0)}_{ij}$ refers to the j-th cluster centroid in solution suggested by the i-th particle. Therefore a swarm suggests a number of candidates for clustering centroids.

Step 2: Evaluate the fitness for each particle based on clustering criteria. The fitness of particle i in swarm is defined as below:

$$F(i) = \frac{\sum_{j=1}^{k} \sum_{\forall yp \in Cij} (yp - zij)^2}{Np}$$

Where N_p is the number of data points as inputs to clustering process. By minimizing the fitness function, the dispersion of clusters would be minimized.

Step 3: If the number of iterations exceeds a predefined level go to Step 7, otherwise go to Step 4.

Step 4: The position of best particle among the particles in swarm is stored. Then the position of all the particles are updated according to Equations (5) and (6). If a particle flies beyond the boundary [X_{min}, X_{max}],(the range of possible solutions) then the position of particle is set to the X_{min} or X_{max}; similarly if a new velocity is beyond the boundary [V_{min}, V_{max}], the new velocity will be set to V_{min} or V_{max} .

Step 5: Reduce the inertia weight - ω - according to the strategy described in Section 3.

Step 6: If the global best of particles, G , remains unchanged for a number of iterations (ten in our implementation) go to Step 7; otherwise go to Step 3.

Step 7: Use the IWC algorithm to finish clustering task. The clustering terminates when one of conditions stated in Section 2 satisfied.

VI. EXPERIMENTS

In order to evaluate the performance of the proposed clustering algorithm, we conducted two experiments using synthetic and real data. In these experiments we compare the proposed PSO-IWC method with PSO clustering, PSO-KM and standalone IWC.

All the experiments are carried out using Matlab R2015a on the same machine with a Core i7 CPU 2.70 GHz, 16.0GB RAM, and Windows 10 operation system.

Figure 2 shows the result of applying PSO-IWC algorithm to the synthetic, Wine and Liver-disorders data sets and shows that the algorithm consistently performs better than the other three approaches even executing many times, so we can see that there are the same two clusters have resulted by applying the algorithm two times. While Figure 3 summarizes the result of applying PSO-KM four times and ensure that the results are differ for each execution. The second experiment was conducted using Iris and Cancer datasets. These data sets are very classical and often used to examine and compare the performances of algorithms in the fields of classification.

The dataset (*) is available online. The second and the third columns of Table 1 show the number of data points in each dataset and in each individual cluster respectively.

The results of clustering on these datasets using the proposed hybrid PSO-IWC, PSO and PSO-KM are presented in Table 2, and here we can see that the results obtained from proposed algorithm is significantly better than the other three approaches, the comparative analysis for different attributes like time, accuracy, error rate and number of iterations are tabulated in Table 3 and the results show a general improvement of performance when using PSO-IWC.

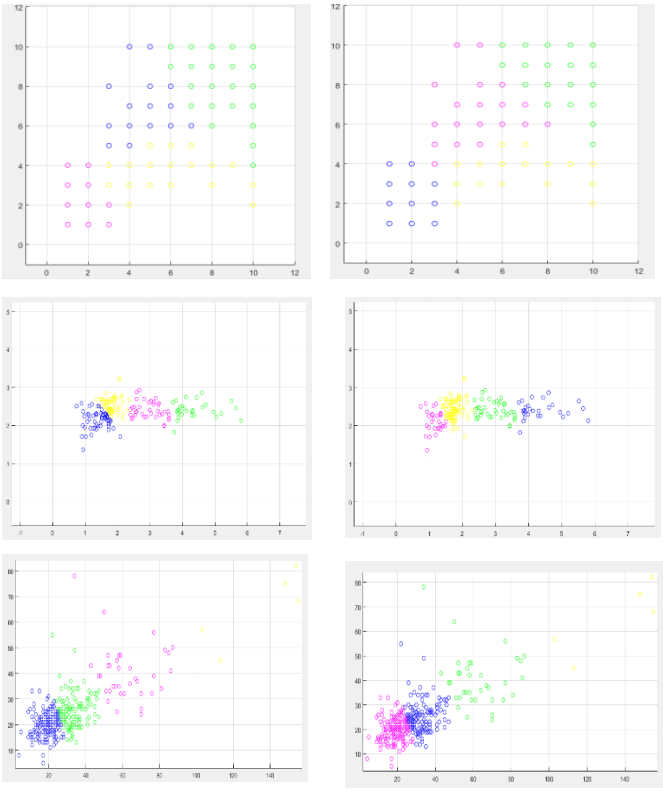

Figure 2. Top: Results of applying PSO-IWC algorithm on artificial data set, Middle: Results of applying PSO-IWC algorithm on Wine data set, Bottom: Results of applying PSO-IWC algorithm on liver-disorders data set

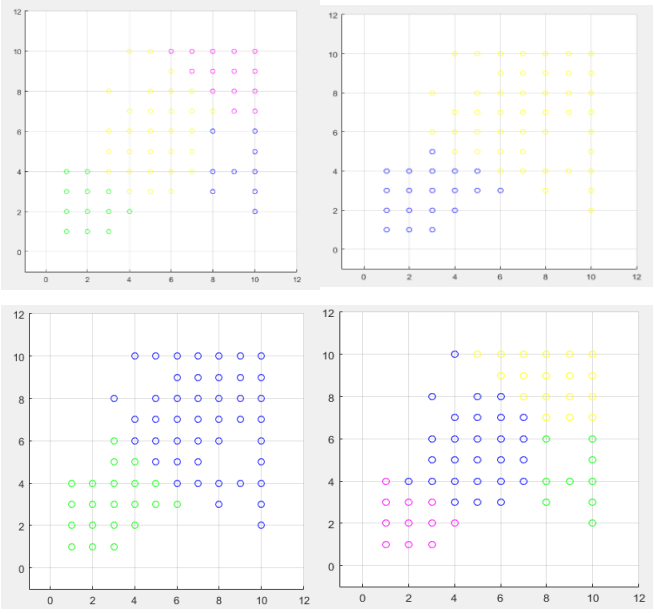

Figure 3. Result of applying PSO-KM algorithm four times (different clusters distribution obtained).

TABLE 1

(INFORMATION FOR DATASETS)

Data Set	# Data in set	# Data in Clusters	# Clusters	# Space dimension
SET I	210	Each 70	3	2
SET II	210	Each 70	3	2
Iris	150	Each 50	3	4
Cancer	683	444 & 239	2	9
Wine	10782	59, 71, 48	3	13
liver-disorders	7297	Each 21	7	7

TABLE 2

INFORMATION FOR SYNTHETIC AND REAL DATASETS

Data Set	Criteria	PSO	PSO-KM	PSO-IWC
SET I	Error rate	0%	0%	0%
SET II	Error rate	7.6%	7.2%	7.01%
Iris	Error rate	12%	12%	10.5%
Cancer	Error rate	4.7%	3.7%	2.87%
Wine	Error rate	6.4%	4.5	3.2
liver-disorders	Error rate	5.02%	8.1%	6.3%

TABLE 3

THE PERFORMANCE OF THREE CLUSTERING METHOD

Data Set	Time (seconds)		Iterations		Accuracy	
	PSO-IWC	PSO-KM	PSO-IWC	PSO-KM	PSO-IWC	PSO-KM
SET I	0.32	0.47	3	4	93.03 %	92.11%
SET II	0.58	0.78	2	3	91.11 %	91.15%
Iris	0.3347	1.7830	2	8	90.69	84.9
Cancer	1.2699	5.3059	3	14	89.7	87.1
Wine	0.3695	1.8361	7	18	92.2%	88.05%
liver-disorders	0.6473	3.7687	8	14	91.36 %	90.06%

VII. CONCLUSIONS

In this paper, we have proposed a method based on combination of the particle swarm optimization (PSO) and the IWC algorithm. We showed that the combined method has the advantage of both PSO and IWC methods. As the PSO algorithm successfully searches all space during the initial stages of a global search, we used PSO algorithm at earlier stage of PSO-IWC. As long as the particles in swarm being close to the global optimum, the algorithm switches to IWC as it can converge faster than PSO algorithm. We detected the proper stage for switching from PSO to IWC using the fitness function.

Future studies will extend the fitness function to also explicitly optimize the higher dimensional problems-and large number of patterns. The PSO-IWC clustering algorithms will also be extended to dynamically determine the optimal number of clusters.

REFERENCES

[1] DW van der Merwe and AP Engelhrecht, Data Clustering using Particle Swarm Optimization. University of Pretoria: South Africa, 2013.

[2] Chen, C.-Y., and Ye, F, " Particle swarm optimization algorithm and its application to clustering analysis," in Proc. the IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan , pp. 789–794,2004.

[3] Paterlini, S., and Krink, T., "Differential evolution and particle swarm optimization in partitional clustering" in Proc. Computational Statistics and Data Analysis, 50, pp 1220–1247. 2006

[4] D.W. Boeringer, and D.H. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," IEEE Transaction of Antennas Propagation 52 (3) pp 771–779. 2004

[5] Wesam Barbakh and Colin Fyfe, "Inverse Weighted Clustering Algorithm", IEEE Transaction of Antennas Propagation 52 (3) pp 771–779. 2004

[6] W. Barbakh. The family of inverse exponential k-means algorithms. Computing and Information Systems, 11(1):1–10, ISSN 1352-9404. 2007

[7] W. Barbakh, M. Crowe, and C. Fyfe. A family of novel clustering algorithms. In 7th international conference on intelligent data engineering and automated learning, IDEAL2006, pages 283–290,. ISSN 0302- 9743 ISBN-13 978-3-540-45485-4. 2006

[8] Shafiq Alam, Gillian Dobbie, Patricia Riddle, "An Evolutionary Particle Swarm Optimization Algorithm for Data Clustering", Swarm Intelligence Symposium St. Louis MO USA, September 21-23, IEEE 2008.

[9] Chunqin Gu, Qian Tao, " Clustering Algorithm Combining CPSO with K-Means", Communication and Knowledge (ICTCK), International Congress on. 2015

[10] Lekshmy P Chandran,K A Abdul Nazeer, “An Improved Clustering Algorithm based on K-Means and Harmony Search Optimization”, IEEE 2011.

[11] D. J. MacKay. Information Theory, Inference, and Learning Algorithms. Cambridge University Press., 2003.

[12] W. Barbakh and C. Fyfe. Inverse weighted clustering algorithm. Computing and Information Systems, 11(2):10–18, ISSN 1352-9404. 2007

[13] J Kennedy, and RC Eberhart, "Particle Swarm Optimization," in Proc. the IEEE International Joint Conference on Neural Networks, Vol. 4, pp 1942–1948, 1995.

[14] Y. Shi, and R.C. Eberhart, "A modified particle swarm optimizer," in Proc. IEEE World Conf. on Computation Intelligence (1998) pp 69–73.

[15] R.C. Eberhart, and Y. Shi, "Comparing Inertia Weights and Constriction Factors in Particle swarm Optimization," in Proc. Congress on Evolutionary Computing, vol. 1 (2000) pp 84–88.

[16] Alireza Ahmadyfard, Hamidreza Modares, " Combining PSO and k means to Enhance Data Clustering," in Telecommunications, 2008.