Data Clustering Using Hybrid Particle Swarm Optimization

Ahmed A.A. Esmin[1,2] and Stan Matwin[2,3]

[1] Department of Computer Science University of Lavras (UFLA),
Lavras, MG, 37200-000, Brazil
ahmed@dcc.ufla.br
[2] School of Electrical Engineering and Computer Science,
University of Ottawa (uOttawa), Ottawa, ON, Canada
[3] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
stan@eecs.uottawa.ca

Abstract. Clustering is an important data mining task and has been explored extensively by a number of researchers for different application areas, such as text application and bioinformatics data. In this paper we propose the use of a novel algorithm for clustering data that we call hybrid particle swarm optimization with mutation (HPSOM), which is based on PSO. The HPSOM basically uses PSO and incorporates the mutation process often used in GA to allow the search to escape from local optima. It is shown how the PSO/HPSOM can be used to find the centroids of a user-specified number of clusters. The new algorithm is evaluated on five benchmark data sets. The proposed method is compared with the K-means (KM) clustering technique and the standard PSO algorithm. The results show that the algorithm is efficient and produces compact clusters.

Keywords: Data Cluster, PSO, Hybrid PSO, Data Mining.

1 Introduction

Clustering is an important problem that often must be solved as part of more complicated tasks in pattern recognition, image analysis, and other fields of science and engineering. Clustering is one of the main tasks in knowledge discovery from databases (KDD) and consists in finding groups within a certain set of data, where each group contains objects similar to each other and different from those of other groups [1].

In the clustering process, the learning algorithm is provided with just the data points and no labels; the task is to find a suitable representation of the underlying distribution of the data (data vectors are grouped based on distance from one to another). Some approaches are based on hybridization of different clustering techniques and involve optimization in the process.

K-means (KM) algorithm is one of the most popular and widespread partitioning clustering algorithms because of its superior feasibility and efficiency in dealing with a large amount of data. The main drawback of the KM algorithm is that the cluster result is sensitive to the selection of the initial cluster centers and may converge to the local optima [2,3].

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 159–166, 2012.
© Springer-Verlag Berlin Heidelberg 2012

The particle swarm optimization (PSO) algorithm is an optimization method developed by Eberhart et al. [4,5]. PSO tries to find the optimal solution through the simulation of some ideas drawn from fish schooling, bird flocking, and other social groups. One such idea is that an agent can effectively achieve his objective using the information that is owned by him and the information that is shared among the group. This means that PSO is an optimization method that uses the principles of social behavior. PSO has proved to be competitive with genetic algorithms in several tasks, mainly in optimization areas [5,6].

PSO has been successfully applied in several areas such as clustering problem [2,3], function optimization [6,7] etc. PSO finds the best value with interaction of particle, solves the problem of initialization of the KM algorithm, but it also can trapped in local optima [6,12].

Different variants of the PSO algorithm have been proposed. Some of these variants have been proposed to incorporate the capabilities of other evolutionary algorithms, such as hybrid versions of PSO or the adaptation of PSO parameters, creating the adaptive PSO versions. Many authors have considered incorporating selection, mutation, and crossover, as well as differential evolution, into the PSO algorithm. As a result, hybrid versions of PSO have been created and tested, including a hybrid of genetic algorithm and PSO (GA-PSO), evolutionary PSO (EPSO) [6-10] and hybrid particle swarm optimization with mutation (HPSOM) algorithm [6,7].

In this paper we explore the HPSOM algorithm to solve the PSO stagnation problem and to prevent the particles from being trapped in local minima [6,7]. The main contribution of this paper is to describe a strategy for cluster data by using the HPSOM algorithm and comparing its results with those obtained by KM and standard PSO. Experimental results indicate the superiority of the HPSOM algorithm.

The rest of the paper is organized as follows: Section 2 provides an overview of PSO, Section 3 presents the HPSOM algorithm, Section 4 introduces the HPSOM clustering algorithm, and Section 5 shows the tests performed with the different variants of the algorithm. The conclusions are presented in Section 6.

2 An Overview of Particle Swarm Optimization

The particle swarm optimization algorithm (PSO) is a population-based optimization method that tries to find the optimal solution using a population of particles [4,5]. Each particle is an individual, and the swarm is composed of particles. In PSO, the solution space of the problem is formulated as a search space. Each position in the search space is a potential solution of the problem. Particles cooperate to find the best position (best solution) in the search space (solution space). Each particle moves according to its velocity. At each iteration, the particle movement is computed as follows:

$$x_i(t+1) \leftarrow x_i(t) + v_1(t), \tag{1}$$

$$v_i(t+1) \leftarrow \omega v_i(t) + c_1 r_1 (pbest_i(t) - x_i(t)) + c_2 r_2 (gbest(t) - x_i(t)) \tag{2}$$

In Eqs. (1), (2), $x_i(t)$ is the position of particle i at time t, $v_i(t)$ is the velocity of particle i at time t, $pbest_i(t)$ is the best position found by particle itself so far, $gbest(t)$ is the best position found by the whole swarm so far, ω is an inertia weight scaling the previous time step velocity, c_1 and c_2 are two acceleration coefficients that scale the influence of the best personal position of the particle ($pbest_i(t)$) and the best global position ($gbest(t)$), r_1 and r_2 are random variables within the range [0,1]. The process of PSO is shown as Fig. 1.

```
Initialize a population of particles with random
positions and velocities in the search space.
While (termination conditions are not met)
{
   For each particle i do
   {
      Update the position of particle i according to
      equation (1).
      Update the velocity of particle i according to
      equation (2).
      Map the position of particle i in the solution
      space and evaluate its fitness value according to
      the fitness function.
      Update pbest_i(t) and gbest_i(t) if necessary.
         }
  }
```

Fig. 1. The process of the PSO algorithm

3 The Hybrid PSO with Mutation Algorithm

Since the presentation of PSO [4,5], its performance has been investigated in several papers. The work presented in [11] describes the complex task of parameter selection in the PSO model. Comparisons between PSOs and the standard genetic algorithm (GA) formulation have been carried out in [11], where the author points out that PSO performs well in the early iterations but presents problems in reaching a near-optimal solution.

The behavior of PSO in the *gbest* model presents some important aspects related to the velocity update. If a particle's current position coincides with the global best position, the particle will only move away from this point if its inertia weigh (ω) and previous velocity are different from zero. If their previous velocities are very close to zero, then all the particles will stop moving once they catch up with the global best particle, which may lead to a premature convergence of the algorithm. In fact, this does not even guarantee that the algorithm has converged on a local minimum. It means that all the particles have converged at the best position discovered so far by

the swarm. This phenomenon is known as stagnation [12]. The solution presented in [12] is based on adding a new parameter and additional equations. Another solution is presented in [13] by introducing *breeding and subpopulation*.

In [6] we proposed hybrid particle swarm optimization with mutation (HPSOM) by incorporating the mutation process often used in GA into PSO. The stagnation is alleviated by this technique and introduces diversity into the population. This process allows the swarm to escape from local optima and to search in different zones of the search space.

This process starts with the random choice of a particle in the swarm and moves to different positions inside the search area. The mutation process is implemented by the following equation (3):

$$mut(p_k) \leftarrow -p_k + \beta \tag{3}$$

where, p_k is the random choice *kth* particles from the swarm, and β is randomly obtained within the range $[0, 0.1*(x_{\max} - x_{\min})]$, representing 0.1 times the length of the search space. Comparisons between standard PSO and HPSOM, which show the HPSOM model as better than the standard PSO model, are presented in [6,7].

4 PSO/HPSOM Clustering

Among all the efforts in the literature to modify the particle swarm optimization algorithm for data clustering, [14-15] seem to be the ones closest to the original idea of PSO since each particle comprehends a whole candidate solution to the problem. A particle p_i is constructed as follows:

$$p_i = (m_{i1}, m_{i2}, \ldots, m_{ij}, \ldots, m_{iNc})$$

where Nc is the number of clusters to be formed, and m_{ij} corresponds to the jth centroid of the ith particle, the centroid of the cluster C_{ij}. Thus, a single particle represents a candidate solution to a given clustering problem.

Each particle is evaluated using the following equation (fitness function):

$$f = \frac{\sum_{j=1}^{Nc} [\sum_{\forall X_k \in C_{ij}} d(x_k, m_{ij}) / | C_{ij} |]}{Nc} \tag{4}$$

where x_k denotes the k^{th} data vector, $| C_{ij} |$ is the number of data vectors belonging to the cluster C_{ij}, and d is the Euclidian distance between x_k and m_{ij}.

The stopping criterion (termination conditions) mentioned in the algorithm depends on the type of problem being solved. Usually, the algorithm is run for a fixed number of iterations (objective function evaluations) or until a specified error bound is reached. In this study, the algorithm is stopped when a user-specified number of iterations has been exceeded. The proposed cluster algorithm is shown in Fig 2:

```
Initialize the cluster centroids of each particle randomly  //allocate data to each particle randomly with the centroid vector values
Repeat
 { For each particle i do
   { For each data vector x_k do
     {Calculate the d(x_k, m_ij) to all cluster centroids m_ij
       Assign x_k to cluster C_ij such that:
                  d(x_k, m_ij) = min ∀l = 1,..., Nc {d(xk, mil)}
                   //assign x_k to the cluster C_ij with the minimum distance
                   // (from all the Nc clusters of the particle_i)
      }
   }
 Calculate the fitness using equation (4)
 Update the swarm particles (centroids) as in (Fig 1)
 Execute mutation process (for HPSOM) using equation (3)
Until a stopping criterion is satisfied.
```

Fig. 2. The cluster HPSOM algorithm

5 Results and Discussion

To evaluate the performance of the proposed algorithm, five benchmarks were used: iris, wine, glass and breast cancer, taken from the UCI Repository of Machine Learning Databases [16] and a synthetic data sets (artificial problem). The main features of these benchmarks are presented as follows:

- **Artificial problem:** This problem follows the following classification rule:

$$class = \begin{cases} 1 & if(z_1 \geq 0.7) or((z_1 \leq 0.3) and(z_2 \geq -0.2 - z_1)) \\ 0 & otherwise \end{cases} \quad (5)$$

A total of 500 data vectors were randomly created, with $z_1, z_2 \sim U(-1,1)$.

- **Iris:** This is a well-understood database with 4 inputs (attributes), 3 classes and 150 data vectors (instances).
- **Wine:** This is a classification problem with well-behaved class structures. There are 13 inputs, 3 classes and 178 data vectors.
- **Glass:** This is a classification problem with 9 inputs, 7 classes and 214 data vectors.
- **Breast cancer:** The Wisconsin breast cancer database contains 9 attributes, 2 classes and 286 data vector. The objective is to classify each data vector into benign or malignant tumors.

For each data set, PSO/HPSOM was run 30 times, with 20 iteration, 10 particles, and the parameters ω = 0.72, c_1 = 1.49, and c_2 = 1.49, after several simulations these

parameters ensures a good convergence. The relatively small number of iterations (function evaluations) was chosen due to the high convergence rate of PSO/HPSOM. HPSOM has an additional parameter related to the mutation rate, which was set to 10%. Each benchmark class was represented by the cluster (of the best particle) with the largest number of data of that class; data of different classes within this cluster were considered misclassified. Other measures that can be used to evaluate the performance are the intra-cluster and the intercluster distance.

The intra-cluster distance measures the density of the created clusters, i.e., how compact these clusters are, since the data in the same cluster should be similar. In this work, the intra-cluster distance was measured by the average distances between data in the same cluster. The intercluster distance measures the separation between created clusters, given that the clusters should be as far as possible from each other. Here the intercluster distance was measured by the average distances between the mass centers of the clusters.

Table 1. Comparison of the results by fitness, correctly cluster instances, intra- and intercluster distance

Prob.	Algo.	Fitness Equation	Correctly clustered (%)	Average Intra-cluster	Average Inter-cluster
Iris	KM	0.0842± 0.0035	81.464 ± 6.6106	3.3126±0.247	0.8981±0.092
	PSO	0.0869±0.00484	79.234 ± 6.9871	3.8954±0.183	0.8915±0.87
	HPSOM	**0.08203±0.00289**	**86.037± 5.0426**	**3.0727±0.178**	**0.8532±0.096**
Wine	KM	0.06155± 0.00145	71.217± 0.5254	4.443±0.265	**1.156 ± 0.14**
	PSO	0.05903± 0.00153	68.712± 2.2641	5.143±0.156	2.989+0.203
	HPSOM	**0.00289±0.00148**	**73.872± 0.5725**	**4.185**±0.132	2.789 ±0.187
Glass	KM	0.01502 ±.00260	41.025± 3.7600	1.7903±0.143	3.8945±0.237
	PSO	0.01911± 0.00108	42.205± 5.3687	1.8353±0.129	**3.4551**±0.157
	HPSOM	**0.01442±0.00123**	**44.108 ±4.6933**	**1.6264**±0.121	5.2453±0.109
Breast-Canc.	KM	1.989± 0.064	71.402± 3.013	6.981± 0.324	**1.986 ±0.252**
	PSO	2.606± 0.084	65.140± 4.413	7.571±0.343	3.443±0.216
	HPSOM	**1.795± 0.139**	**73.230 ±5.573**	6.752±0.402	3.295±0.96
Artif.	KM	0.997±0.042	51.183 ± 5.103	3.678±0.087	1.83±0.044
	PSO	0.781±0.028	54.174 ±6.265	3.826±0.89	1.I92±0.51
	HPSOM	**0.772±0.027**	**57.174 ±5.662**	**3.801±0.81**	**1.160 ±0.43**

Table 1 summarize the results obtained from the three clustering algorithms for the benchmark problems above. The values reported are the averages from over 30 simulations, with the standard deviations indicating the range of values at which the algorithms converge. First, consider the fitness of solutions, i.e., the equation (4). For all the problems, the hybrid algorithm had the smallest average quantization error (fitness functions). For the iris and glass problems, KM clustering was not significantly worse than the PSO and HPSOM algorithms (the difference in not high).

However, for the wine problem both KM and the PSO algorithm were significantly worse than the hybrid algorithm (HPSOM is 2 to 3 times better).

When considering the inter- and intra-cluster distances, the latter ensures compact clusters with little deviation from the cluster centroids, while the former ensures larger separation between the different clusters. With reference to these criteria, PSO approaches succeeded most in finding clusters with larger separation than did the KM algorithm; the HPSOM algorithm succeeded in four of the five problems. HPSOM formed the most compact clusters for the five problems. Figures 3 illustrate an example of the clustering found.

Figure 4 summarizes the effect of varying the number of clusters for the three algorithms for the artificial problem. In this case, it is expected that the quantization error should go down when the number of clusters increases. Figure 4 also shows that the HPSOM algorithm consistently performs better than the other two algorithms when the number of clusters increases.

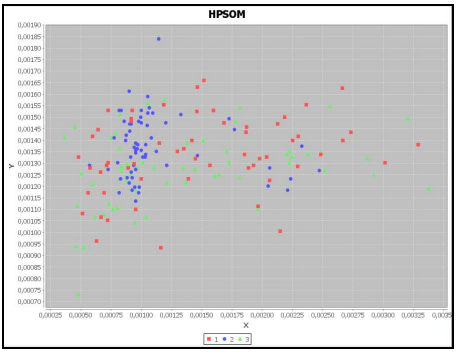

Fig. 3. Clustering found HPSOM Wine

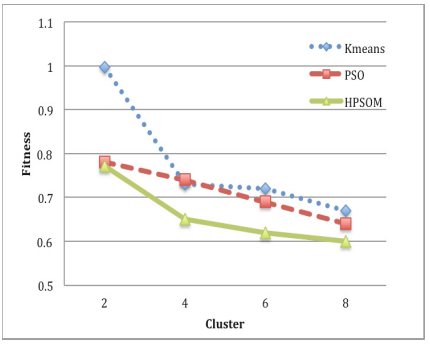

Fig. 4. Effect of the different number of clusters on artificial problem

6 Conclusion

This paper investigates the application of a hybrid PSO called HPSOM to cluster data vectors. Two algorithms were tested, namely, standard PSO and a hybrid PSO approach where the individuals of the swarm are seeded by the result of the KM algorithm. Comparison of the two PSO approaches with KM clustering showed that HPSOM algorithm approaches have better convergence with lower fitness errors and, in general, larger intercluster and smaller intra-cluster distances.

Future works will extend the fitness function to also explicitly optimize the intercluster and intra-cluster distances, and work toward the development of a hybrid PSO clustering method capable of handling large and complex data set and determining the optimal number of clusters.

Acknowledgements. We would like to thank CNPq, FAPEMIG (Brazilian agencies) and NSERC (Canada) for partial financial support. The authors also thank the anonymous reviewers for useful remarks and suggestions.

References

1. Jiawei, H., Micheline, K.: Data Mining, Conecpts and Techniques. Morgan Kaufmann Publishers (2001)
2. Kao, Y.T., Zahara, E., Kao, I.W.: A hybridized approach to data clustering. Expert Systems with Applications 34(3), 1754–1762 (2008)
3. Feng, H.M., Chen, C.Y., Ye, F.: Evolutionary fuzzy particle swarmoptimization vector quantization learning scheme in image compression. Expert Systems with Applications 32(1), 213–222 (2007)
4. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of IEEE Internal Conference on Neural Networks, Perth, Australia, vol. 4, pp. 1942–1948 (1995)
5. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39–43 (1995)
6. Esmin, A.A.A., Lambert-Torres, G., Zambroni de Souza, A.C.: A Hybrid Particle Swarm Optimization Applied to Loss Power Minimization. IEEE Transactions on Power Systems 20(2), 859–866 (2005)
7. Esmin, A.A.A., Lambert-Torres, G.: Fitting Fuzzy Membership Functions using Hybrid Particle Swarm Optimization. In: 2006 IEEE World Congress on Computational Intelligence & IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2006, pp. 9954–9961. IEEE Press, Vancouver (2006)
8. Miranda, V., Fonseca, N.: EPSO-evolutionary particle swarm optimization, a new algorithm with applications in power systems. In: Proc. of the Asia Pacific IEEE/PES Transmission and Distribution Conference and Exhibition, vol. 2, pp. 745–750 (2002)
9. Fan, S.S., Liang, Y., Zahara, E.: Hybrid simplex search and particle swarm optimization for the global optimization of multimodal functions. Engineering Optimization 36(4), 401–418 (2004)
10. Shi, Y., Eberhart, R.: Parameter Selection in Particle Swarm Optimization. In: Proc. 7th Annual Conference on Evolutionary Programming, pp. 591–600 (1998)
11. Angeline, P.: Evolutionary Optimization Versus Particle Swarm Optimization Philosophy and Performance Differences. In: Proc. 7th Annual Conference on Evolutionary Programming, pp. 601–610 (1998)
12. Van den Bergh, F., Engelbrecht, A.P.: A New Locally Convergent Particle Swarm Optimiser. In: Proc. of the IEEE International Conference on Systems, Man and Cybernetics, Hammamet, Tunisia (October 2002)
13. Løvbjerg, M., Rasmussen, T.K., Krink, T.: Hybrid Particle Swarm Optimiser with Breeding and Subpopulations. In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO), USA (July 2001)
14. van der Merwe, D.W., Engelbrecht, A.P.: Data Clustering using Particle Swarm Optimization. In: Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003), pp. 215–220. IEEE Computer Society, Caribella (2003)
15. Esmin, A.A.A., Pereira, D.L., Araujo, F.P.A.: Study of Different Approach to Clustering Data by Using The Particle Swarm Optimization Algorithm. In: 2008 IEEE Congress on Evolutionary Computation (IEEE CEC 2008), Proceedings of IEEE Congress on Evolutionary Computation, Hong Kong, pp. 1817–1822 (2008)
16. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, Department of Information and Computer Science, Irvine (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`