Data Clustering using Particle Swarm Optimization

DW van der Merwe
Department of Computer Science
University of Pretoria
tjippie@fouriersystems.co.za

AP Engelbrecht
Department of Computer Science
University of Pretoria
engel@driesie.cs.up.ac.za

Abstract- This paper proposes two new approaches to using PSO to cluster data. It is shown how PSO can be used to find the centroids of a user specified number of clusters. The algorithm is then extended to use K-means clustering to seed the initial swarm. This second algorithm basically uses PSO to refine the clusters formed by K-means. The new PSO algorithms are evaluated on six data sets, and compared to the performance of K-means clustering. Results show that both PSO clustering techniques have much potential.

1 Introduction

Data clustering is the process of grouping together similar multi-dimensional data vectors into a number of clusters or bins. Clustering algorithms have been applied to a wide range of problems, including exploratory data analysis, data mining [4], image segmentation [12] and mathematical programming [1, 16]. Clustering techniques have been used successfully to address the scalability problem of machine learning and data mining algorithms, where prior to, and during training, training data is clustered, and samples from these clusters are selected for training, thereby reducing the computational complexity of the training process, and even improving generalization performance [6, 15, 14, 3].

Clustering algorithms can be grouped into two main classes of algorithms, namely supervised and unsupervised. With supervised clustering, the learning algorithm has an external teacher that indicates the target class to which a data vector should belong. For unsupervised clustering, a teacher does not exist, and data vectors are grouped based on distance from one another. This paper focuses on unsupervised clustering.

Many unsupervised clustering algorithms have been developed. Most of these algorithms group data into clusters independent of the topology of input space. These algorithms include, among others, K-means [7, 8], ISODATA [2], and learning vector quantizers (LVQ) [5]. The self-organizing feature map (SOM) [11], on the other hand, performs a topological clustering, where the topology of the original input space is maintained. While clustering algorithms are usually supervised or unsupervised, efficient hybrids have been developed that performs both supervised and unsupervised learning, e.g. LVQ-II [5].

Recently, particle swarm optimization (PSO) [9, 10] has been applied to image clustering [13]. This paper explores the applicability of PSO to cluster data vectors. In the process of doing so, the objective of the paper is twofold:

- to show that the standard PSO algorithm can be used to cluster arbitrary data, and
- to develop a new PSO-based clustering algorithm where K-means clustering is used to seed the initial swarm.

The rest of the paper is organized as follows: Section 2 presents an overview of the K-means algorithm. PSO is overviewed in section 3. The two PSO clustering techniques are discussed in section 4. Experimental results are summarized in section 5.

2 K-Means 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. K-means clustering groups data vectors into a predefined number of clusters, based on Euclidean distance as similarity measure. Data vectors within a cluster have small Euclidean distances from one another, and are associated with one centroid vector, which represents the "midpoint" of that cluster. The centroid vector is the mean of the data vectors that belong to the corresponding cluster.

For the purpose of this paper, define the following symbols:

- N_d denotes the input dimension, i.e. the number of parameters of each data vector
- N_o denotes the number of data vectors to be clustered
- N_c denotes the number of cluster centroids (as provided by the user), i.e. the number of clusters to be formed
- $\mathbf{z}_p$ denotes the p-th data vector
- $\mathbf{m}_j$ denotes the centroid vector of cluster j

0-7803-7804-0 /03/$17.00 © 2003 IEEE

- n_j is the number of data vectors in cluster j
- C_j is the subset of data vectors that form cluster j.

Using the above notation, the standard K-means algorithm is summarized as

1. Randomly initialize the N_c cluster centroid vectors
2. **Repeat**
   (a) For each data vector, assign the vector to the class with the closest centroid vector, where the distance to the centroid is determined using

$$d(\mathbf{z}_p, \mathbf{m}_j) = \sqrt{\sum_{k=1}^{N_d} (z_{pk} - m_{jk})^2} \quad (1)$$

   where k subscripts the dimension.
   (b) Recalculate the cluster centroid vectors, using

$$\mathbf{m}_j = \frac{1}{n_j} \sum_{\forall \mathbf{z}_p \in C_j} \mathbf{z}_p \quad (2)$$

   until a stopping criterion is satisfied.

The K-means clustering process can be stopped when any one of the following criteria are satisfied: when the maximum number of iterations has been exceeded, when there is little change in the centroid vectors over a number of iterations, or when there are no cluster membership changes. For the purposes of this study, the algorithm is stopped when a user-specified number of iterations has been exceeded.

3 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based stochastic search process, modeled after the social behavior of a bird flock [9, 10]. The algorithm maintains a population of particles, where each particle represents a potential solution to an optimisation problem.

In the context of PSO, a swarm refers to a number of potential solutions to the optimization problem, where each potential solution is referred to as a particle. The aim of the PSO is to find the particle position that results in the best evaluation of a given fitness (objective) function.

Each particle represents a position in N_d dimensional space, and is "flown" through this multi-dimensional search space, adjusting its position toward both

- the particle's best position found thus far, and
- the best position in the neighborhood of that particle.

Each particle i maintains the following information:

- $\mathbf{x}_i$: The *current position* of the particle;
- $\mathbf{v}_i$: The *current velocity* of the particle;
- $\mathbf{y}_i$: The *personal best position* of the particle.

Using the above notation, a particle's position is adjusted according to

$$\begin{aligned} v_{i,k}(t+1) &= w v_{i,k}(t) + c_1 r_{1,k}(t)(y_{i,k}(t) - x_{i,k}(t)) + \\ &\quad c_2 r_{2,k}(t)(\hat{y}_k(t) - x_{i,k}(t)) \quad (3) \\ \mathbf{x}_i(t+1) &= \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (4) \end{aligned}$$

where w is the inertia weight, c_1 and c_2 are the acceleration constants, $r_{1,j}(t), r_{2,j}(t) \sim U(0,1)$, and $k = 1, \cdots, N_d$. The velocity is thus calculated based on three contributions: (1) a fraction of the previous velocity, (2) the cognitive component which is a function of the distance of the particle from its personal best position, and (3) the social component which is a function of the distance of the particle from the best particle found thus far (i.e. the best of the personal bests).

The personal best position of particle i is calculated as

$$\mathbf{y}_i(t+1) = \begin{cases} \mathbf{y}_i(t) & \text{if } f(\mathbf{x}_i(t+1)) \geq f(\mathbf{y}_i(t)) \\ \mathbf{x}_i(t+1) & \text{if } f(\mathbf{x}_i(t+1)) < f(\mathbf{y}_i(t)) \end{cases} \quad (5)$$

Two basic approaches to PSO exists based on the interpretation of the neighborhood of particles. Equation (3) reflects the *gbest* version of PSO where, for each particle, the neighborhood is simply the entire swarm. The social component then causes particles to be drawn toward the best particle in the swarm. In the *lbest* PSO model, the swarm is divided into overlapping neighborhoods, and the best particle of each neighborhood is determined. For the *lbest* PSO model, the social component of equation (3) changes to

$$c_2 r_{2,k}(t)(\hat{y}_{j,k}(t) - x_{i,k}(t)) \quad (6)$$

where $\hat{\mathbf{y}}_j$ is the best particle in the neighborhood of the i-th particle.

The PSO is usually executed with repeated application of equations (3) and (4) until a specified number of iterations has been exceeded. Alternatively, the algorithm can be terminated when the velocity updates are close to zero over a number of iterations.

4 PSO Clustering

In the context of clustering, a single particle represents the N_c cluster centroid vectors. That is, each particle $\mathbf{x}_i$ is constructed as follows:

$$\mathbf{x}_i = (\mathbf{m}_{i1}, \cdots, \mathbf{m}_{ij}, \cdots, \mathbf{m}_{iN_c}) \quad (7)$$

where $\mathbf{m}_{ij}$ refers to the j-th cluster centroid vector of the i-th particle in cluster C_{ij}. Therefore, a swarm represents a number of candidate clusterings for the current data vectors. The fitness of particles is easily measured as the quantization error,

$$J_e = \frac{\sum_{j=1}^{N_c}[\sum_{\forall \mathbf{z}_p \in C_{ij}} d(\mathbf{z}_p, \mathbf{m}_j)/|C_{ij}|]}{N_c} \quad (8)$$

where d is defined in equation (1), and $|C_{ij}|$ is the number of data vectors belonging to cluster C_{ij}, i.e. the frequency of that cluster.

This section first presents a standard *gbest* PSO for clustering data into a given number of clusters in section 4.1, and then shows how K-means and the PSO algorithm can be combined to further improve the performance of the PSO clustering algorithm in section 4.2.

4.1 *gbest* PSO Cluster Algorithm

Using the standard *gbest* PSO, data vectors can be clustered as follows:

1. Initialize each particle to contain N_c randomly selected cluster centroids.
2. For $t = 1$ to t_{max} do
   (a) For each particle i do
   (b) For each data vector $\mathbf{z}_p$
      i. calculate the Euclidean distance $d(\mathbf{z}_p, \mathbf{m}_{ij})$ to all cluster centroids C_{ij}
      ii. assign $\mathbf{z}_p$ to cluster C_{ij} such that $d(\mathbf{z}_p, \mathbf{m}_{ij}) = \min_{\forall c=1,\cdots,N_c}\{d(\mathbf{z}_p, \mathbf{m}_{ic})\}$
      iii. calculate the fitness using equation (8)
   (c) Update the global best and local best positions
   (d) Update the cluster centroids using equations (3) and (4).

where t_{max} is the maximum number of iterations.

The population-based search of the PSO algorithm reduces the effect that initial conditions has, as opposed to the K-means algorithm; the search starts from multiple positions in parallel. Section 5 shows that the PSO algorithm performs better than the K-means algorithm in terms of quantization error.

4.2 Hybrid PSO and K-Means Clustering Algorithm

The K-means algorithm tends to converge faster (after less function evaluations) than the PSO, but usually with a less accurate clustering [13]. This section shows that the performance of the PSO clustering algorithm can further be improved by seeding the initial swarm with the result of the K-means algorithm. The hybrid algorithm first executes the K-means algorithm once. In this case the K-means clustering is terminated when (1) the maximum number of iterations is exceeded, or when (2) the average change in centroid vectors is less that 0.0001 (a user specified parameter). The result of the K-means algorithm is then used as one of the particles, while the rest of the swarm is initialized randomly. The *gbest* PSO algorithm as presented above is then executed.

5 Experimental Results

This section compares the results of the K-means, PSO and Hybrid clustering algorithms on six classification problems. The main purpose is to compare the quality of the respective clusterings, where quality is measured according to the following three criteria:

- the quantization error as defined in equation (8);
- the intra-cluster distances, i.e. the distance between data vectors within a cluster, where the objective is to minimize the intra-cluster distances;
- the inter-cluster distances, i.e. the distance between the centroids of the clusters, where the objective is to maximize the distance between clusters.

The latter two objectives respectively correspond to crisp, compact clusters that are well separated.

For all the results reported, averages over 30 simulations are given. All algorithms are run for 1000 function evaluations, and the PSO algorithms used 10 particles. For PSO, $w = 0.72$ and $c_1 = c_2 = 1.49$. These values were chosen to ensure good convergence [17].

The classification problems used for the purpose of this paper are

- **Artificial problem 1**: This problem follows the following classification rule:

$$\text{class} = \begin{cases} 1 & \text{if } (z_1 \geq 0.7) \text{ or } ((z_1 \leq 0.3) \\ & \text{and } (z_2 \geq -0.2 - z_1)) \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

  A total of 400 data vectors were randomly created, with $z_1, z_2 \sim U(-1, 1)$. This problem is illustrated in figure 1.

- **Artificial problem 2**: This is a 2-dimensional problem with 4 unique classes. The problem is interesting in that only one of the inputs are really relevant to the formation of the classes. A total of 600 patterns were drawn from four independent bivariate normal distributions, where classes were distributed according to

$$N_2\left(\mu = \begin{pmatrix} m_i \\ 0 \end{pmatrix}, \sum = \begin{bmatrix} 0.50 & 0.05 \\ 0.05 & 0.50 \end{bmatrix}\right) \quad (10)$$

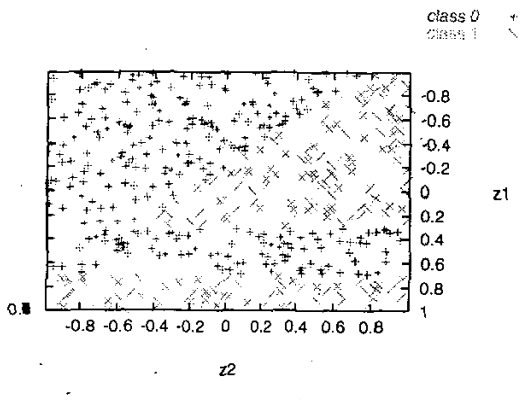

Figure 1: Artificial rule classification problem defined in equation (9)

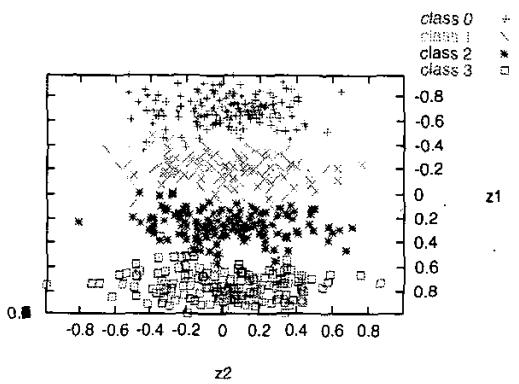

Figure 2: Four-class artificial classification problem defined in equation (10)

for $i = 1, \cdots, 4$, where μ is the mean vector and $\sum$ is the covariance matrix; $m_1 = -3, m_2 = 0, m_3 = 3$ and $m_4 = 6$. The problem is illustrated in figure 2.

- **Iris plants database**: This is a well-understood database with 4 inputs, 3 classes and 150 data vectors.
- **Wine**: This is a classification problem with "well behaved" class structures. There are 13 inputs, 3 classes and 178 data vectors.
- **Breast cancer**: The Wisconsin breast cancer database contains 9 relevant inputs and 2 classes. The objective is to classify each data vector into benign or malignant tumors.
- **Automotives**: This is an 11-dimensional data set representing different attributes of more than 500 automobiles from a car selling agent.

Table 1 summarizes the results obtained from the three clustering algorithms for the problems above. The values reported are averages over 30 simulations, with standard deviations to indicate the range of values to which the algorithms converge. First, consider the fitness of solutions, i.e. the quantization error. For all the problems, except for Artificial 2, the Hybrid algorithm had the smallest average quantization error. For the Artificial 2 problem, the PSO clustering algorithm has a better quantization error, but not significantly better than the Hybrid algorithm. It is only for the Wine and Iris problems that the standard K-means clustering is not significantly worse than the PSO and Hybrid algorithms. However, for the Wine problem, both K-means and the PSO algorithms are significantly worse than the Hybrid algorithm.

When considering 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, the PSO approaches succeeded most in finding clusters with larger separation than the K-means algorithm, with the Hybrid PSO algorithm doing so for 4 of the 6 problems. It is also the PSO approaches that succeeded in forming the more compact clusters. The Hybrid PSO formed the most compact clusters for 4 problems, the standard PSO for 1 problem, and the K-means algorithm for 1 problem.

The results above show a general improvement of performance when the PSO is seeded with the outcome of the K-means algorithm.

Figure 3 summarizes the effect of varying the number of clusters for the different algorithms for the first artificial problem. It is expected that the quantization error should go down with increase in the number of clusters, as illustrated. Figure 3 also shows that the Hybrid PSO algorithm consistently performs better than the other two approaches with an increase in the number of clusters.

Figure 4 illustrates the convergence behavior of the algorithms for the first artificial problem. The K-means algorithm exhibited a faster, but premature convergence to a large quantization error, while the PSO algorithms had slower convergence, but to lower quantization errors. As indicated (refer to the circles) in figure 4, the K-means algorithm converged after 12 function evaluations, the Hybrid

Table 1: Comparison of K-means, PSO and Hybrid clustering algorithms

Problem	Algorithm	*Quantization* Error	*Intra-cluster* Distance	*Inter-cluster* Distance
Artificial 1	K-means	0.984±0.032	3.678±0.085	1.771±0.046
	PSO	0.769±0.031	3.826±0.091	1.142±0.052
	Hybrid	0.768±0.048	3.823±0.081	1.151±0.043
Artificial 2	K-means	0.264±0.001	0.911±0.027	0.796±0.022
	PSO	0.252±0.001	0.873±0.023	0.815±0.019
	Hybrid	0.250±0.001	0.869±0.018	0.814±0.011
Iris	K-means	0.649±0.146	3.374±0.245	0.887±0.091
	PSO	0.774±0.094	3.489±0.186	0.881±0.086
	Hybrid	0.633±0.143	3.304±0.204	0.852±0.097
Wine	K-means	1.139±0.125	4.202±0.223	1.010±0.146
	PSO	1.493±0.095	4.911±0.353	2.977±0.241
	Hybrid	1.078±0.085	4.199±0.514	2.799±0.111
Breast-cancer	K-means	1.999±0.054	6.599±0.332	1.824±0.251
	PSO	2.536±0.197	7.285±0.351	3.545±0.204
	Hybrid	1.890±0.125	6.551±0.436	3.335±0.097
Automotive	K-means	1030.714±44.69	11032.355±342.2	1037.920±22.14
	PSO	971.553±44.11	11675.675±341.1	988.818±22.44
	Hybrid	902.414±43.81	11895.797±340.7	952.892±21.55

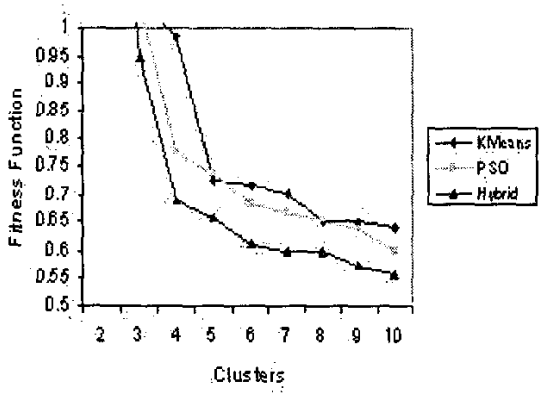

Figure 3: Effect of different number of clusters on Artificial Problem 1

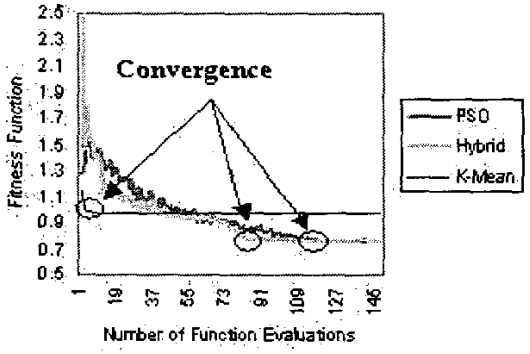

Figure 4: Algorithm convergence for Artificial Problem 1

PSO algorithm after 82 function evaluations, and the standard PSO after 120 function evaluations.

6 Conclusions

This paper investigated the application of the PSO to cluster data vectors. Two algorithms were tested, namely a standard gbest PSO and a Hybrid approach where the individuals of the swarm are seeded by the result of the K-means algorithm. The two PSO approaches were compared against K-means clustering, which showed that the PSO approaches

have better convergence to lower quantization errors, and in general, larger inter-cluster distances and smaller intra-cluster distances.

Future studies will extend the fitness function to also explicitly optimize the inter- and intra-cluster distances. More elaborate tests on higher dimensional problems and large number of patterns will be done. The PSO clustering algorithms will also be extended to dynamically determine the optimal number of clusters.

Bibliography

[1] HC Andrews, "Introduction to Mathematical Techniques in Pattern Recognition", John Wiley & Sons, New York, 1972.

[2] G Ball, D Hall, "A Clustering Technique for Summarizing Multivariate Data", Behavioral Science, Vol. 12, pp 153–155, 1967.

[3] AP Engelbrecht, "Sensitivity Analysis of Multilayer Neural Networks", PhD Thesis, Department of Computer Science, University of Stellenbosch, Stellenbosch, South Africa, 1999.

[4] IE Evangelou, DG Hadjimitsis, AA Lazakidou, C Clayton, "Data Mining and Knowledge Discovery in Complex Image Data using Artificial Neural Networks", Workshop on Complex Reasoning an Geographical Data, Cyprus, 2001.

[5] LV Fausett, "Fundamentals of Neural Networks", Prentice Hall, 1994.

[6] D Fisher, "Knowledge Acquisition via Incremental Conceptual Clustering", Machine Learning, Vol. 2, pp 139-172, 1987.

[7] E Forgy, "Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification", Biometrics, Vol. 21, pp 768–769, 1965.

[8] JA Hartigan, "Clustering Algorithms", John Wiley & Sons, New York, 1975.

[9] J Kennedy, RC Eberhart, "Particle Swarm Optimization", Proceedings of the IEEE International Joint Conference on Neural Networks, Vol. 4, pp 1942–1948, 1995.

[10] J Kennedy, RC Eberhart, Y Shi, "Swarm Intelligence", Morgan Kaufmann, 2002.

[11] T Kohonen, "Self-Organizing Maps", Springer Series in Information Sciences, Vol 30, Springer-Verlag, 1995.

[12] T Lillesand, R Keifer, "Remote Sensing and Image Interpretation", John Wiley & Sons, 1994.

[13] M Omran, A Salman, AP Engelbrecht, "Image Classification using Particle Swarm Optimization", Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, Singapore, 2002.

[14] G Potgieter, "Mining Continuous Classes using Evolutionary Computing", M.Sc Thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa, 2002.

[15] JR Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann, San Mateo, 1993.

[16] MR Rao, "Cluster Analysis and Mathematical Programming", Journal of the American Statistical Association, Vol. 22, pp 622-626, 1971.

[17] F van den Bergh, "An Analysis of Particle Swarm Optimizers", PhD Thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa, 2002.