Swarm Based Clustering Methods Using Ant Colony Optimization and Particle Swarm Optimization

Aiden Bickford aidan.bickford@gmail.com
Nate Norberg natenorberg@gmail.com

Abstract

This paper compares traditional clustering algorithms to swarm based algorithms on a clustering problem. The traditional algorithms used are the k-Means algorithm and a competitive neural network. The swarm based algorithms we are comparing are an adaptation of Any Colony Optimization (ACO) and Particle Swarm Optimization (PCO). For this paper, we created an implementation of these algorithms in Java and evaluated them on various data sets.

1. Introduction

Clustering is a common and valuable problem to solve. It is similar to classification where an algorithm learns to distinguish between a set of different classes or *labels* based on the characteristics of the data sets. With classification, the algorithms are trained with data sets that give the correct class label. This is called *supervised learning* After the algorithm is trained, it can predict future classes based off of what it has already learned.

The difference between classification and clustering is that clustering is an *unsupervised learning* problem. This means that there are no pre-specified class labels for the algorithms to learn. The goal of clustering is to find elements that are similar and group them together into their own class. The effectiveness of the clustering is measured by a combination of measuring how close together elements of the same cluster are (known as *intra-cluster distance*) and how far apart elements of one cluster are from the neighboring clusters, (known as *inter-cluster distance*).

2. Background

2.1 K-Means

K-means is a fairly simple method for clustering. It takes an iterative approach to find the centers of the clusters. To begin, the algorithm randomly generates k points in n dimensional space to be the initial centers of the clusters, where n is the number of features in the input data. From here, each data point is associated with the centroid that it is closest to. Once all of the points have been assigned to a cluster, the centers are recalculated to be the center of gravity for the cluster. To find the new center, all the points that are associated as part of that cluster are averaged and the result becomes the new centroid. From there, the process repeats and each point is assigned to the closest centroid with the new value and the centroid coordinates are updated again. This process is repeated until none of the points change to be part of a different cluster. At this point, the algorithm is said to be converged.

2.2 Competitive Learning

The competitive learning algorithm clusters data with the use of a neural network. This network is very simple, compared to other neural networks such as Multi Layer Perceptrons. It consists of an input layer and an output layer with each node of the input layer connected to each node of the output layer. Each edge in this graph has a weight value, which could also be treated as a coordinate in that specific dimension. All of these weights are randomly assigned at the start of the algorithm. Therefore, the network starts out by representing a set number of randomly generated cluster centers, much like k-means.

This network learns as data is passed through it. Each node in the output layer takes the squared difference between each of the input edges and the associated weights and sums it together. This value represents the difference between the input and the center of the cluster. The output node that has the lowest distance value is deemed the winner and the input point is associated with that cluster. This network is a *winner-take-all* algorithm which means that only one output neuron can fire. Also, only the weights into this node are updated after the network is run. The weights are updated by moving the weights closer to the input values.

$$\Delta w_i = \eta \times (x_i - w_i)$$

This serves to minimize the distance between that point and the center in future runs while also moving the center farther away from less similar points.

2.3 Ant Colony Optimization

Ant colonies have been an inspiring source for novel approaches to problem solving. Models of their complex behavior and organization has shown to be a valuable approach to optimization problems. Using ant colonies to cluster data was inspired by the movement and organization of corpses within the ant colonies, and is known as cemetery organization. The data points are randomly placed on a two dimensional grid, the ants then traverse the grid at random picking up and dropping the data points with a calculated probability. The probability of an ant picking up or dropping an item is based on the similarity of the other data points surrounding it, such that the probability of a point being put down increases as it is brought closer to similar points. As this process is continued the data points migrate with the help of the ants towards clusters of related points.

2.4 Particle Swarm Optimization

Particle swarm algorithms are based off of the movement of birds in flocks, and fish in schools. A population of particles is created where each particle represents a potential solution. The particles then move towards an improved solution through communication with the rest of the swarm of particles. This allows the particles to converge on points informed by their personal best, and the performance of the other particles in the swarm. The algorithm implemented in these experiments communicated on a global level, where all the particles were influenced by the best global individual.

3. Hypothesis

The ant colony clustering algorithm will not perform well on datasets where the classes have a small inter cluster distance. This is because the movement of the particles is based on euclidean distance, and if the distance euclidean between between adjacent points in adjacent clusters is low they could bridge the two clusters, and the ant could falsely cluster them together.

The particle swarm clustering algorithm is not as susceptible to this phenomenon. This is because the particles in the swarm move towards the the configurations with the best fitness which is based on a intra-cluster distance, and intra-cluster distance takes into account the relationship of all the points in the cluster, and doesnt rely on the individual relationships between points. However this works against PSO and K Means in cases where clusters are pushed by local optima to cover regions of dichotomous elements.

The ant colony clusters could also become bifurcated on opposite sides of the map.This could occur if the two clusters are large enough to prevent any of the points from being moved away, and the clusters are far enough apart that the distance between them will not be filled by additional points, but the points within each cluster belong in one cluster.

None of the other clustering algorithms are susceptible to this because in the other algorithms the clusters are created and evaluated on a global level where are data points are placed in the best cluster, and not a cluster that is acceptably similar to the point.

4. Approach

4.1 Data Sets

The data sets used for this paper are taken from the UCI Machine Learning Repository (Bache, 2013). These sets have between 5 and 166 attributes. Because these input values vary across many different ranges, we normalized the values between 0 and 1 based off of the respective minimum and maximum values. This allows the random placement of the centers for the different clustering algorithms to more accurately represent the input space.

4.2 Ant Cemetery Organization

The ant colony clustering algorithm uses cemetery organization to cluster the sets of classification data points. The all of the data points were randomly dispersed on a two dimensional grid which was large enough so that the data points took up approximately half of the space. This size was selected after multiple trials as it allowed enough space for the ants and the particles to move, while at the same time not providing so much space that many small clusters would form.

Ants were placed on the grid at random. An ant population the approximate size of one fifth of the number of the population was used. This provided enough ants to perform the clustering while not so many that too many data points would be picked up at once, which would decrease the density calculation, and lower the probability that the ants would ever drop the point.

The ants moved at random to one of the immediate surrounding points on the grid. If the point was occupied with a datapoint, it was picked up with based on the probability of pickup and putdown at following points based on the probability of dropping the point.(Engelbrecht, 2007)

$$P_{pickup} = (\frac{\gamma_1}{\gamma_1 + \lambda})^2$$

$$P_{drop} = (\frac{lambda}{\gamma_2 + \lambda})^2$$

The term λ represents the density of points nearby which are similar to the the point in question. It is calculated measuring the distance between itself and all of the points within a determined visible range of the ant.(Engelbrecht, 2007)

$$\lambda(y_a) = max\left\{0, \frac{1}{n_{\mathcal{N}}^2} \sum_{y_b \in n_{\mathcal{N}} \times n_{\mathcal{N}}(i)} \left(1 - \frac{d(y_a, y_b)}{\gamma}\right)\right\}$$

where $d(y_a, y_b)$ represents the difference between the two points, computed using the euclidean distance. $n_{\mathcal{N}} \times n_{\mathcal{N}}(i)$ represents the visibility of the ant, or the square of area surrounding the ant that it is aware of. This area was linearly increased as the algorithm progressed from a radius of 1 to 3 or 4. (Engelbrecht, 2007)

$$Visibility_{radius} = (1 - Visibility_{radiusfinal})\frac{t_{final} - t}{t_{final}} + Visibility_{radiusfinal}$$

This allowed for the initial clusters to be formed using a lower visibility which minimizes computational cost, and for the refinement of the clusters towards the end of the run time. The values of γ, γ_1, and γ_2 were determined through trial and error to find the optimal values for each data set.

Dataset	X-Dimension	Y-Dimension	Ant Number	Final Visibility	γ	γ_1	γ_2
Breast Cancer	35	35	100	3	.5	.4	.8
Cardiotocography	65	65	420	4	.9	.5	.8
Climate Model	35	35	100	3	.5	.4	.8
E. Coli	26	26	60	3	.7	.7	.8
Glass	20	20	40	3	.9	.7	.8
Hill Valley	50	50	220	3	.9	.7	.8
Liver	34	34	120	3	.9	.7	.8
Ionosphere	27	27	70	3	.9	.5	.8
Knowlege	23	23	50	3	.7	.5	.8
Fertility	15	15	20	3	.9	.5	.8

The clusters were removed from the mapping created by the ant using an agglomerative hierarchical clustering algorithm. The algorithm designates each point on the map as its own cluster. Then the two closest clusters are combined decreasing the number of clusters by one. This process is repeated until the target number of clusters has been found. (Handl, 2003)

4.3 Particle Swarm Optimization

The particle swarm optimization algorithm used a *gbest* configuration. Each particle maintained a group of centroids which represented the current possible clustering of the of the data set, in addition to the personal best centroids for that cluster, and the current velocity for each centroid.. The dimensions of the centroids were initiated to random values between 1 and 0. Then each time period each particle evaluated the clustering ability of it centroids.

$$f = \frac{\sum_{j=1}^{N_c}\left[\sum_{\forall Z_p \in C_i} d(z_p, m_j)\right] |C_{ij}|}{N_c}$$

(Engelbrecht, 2007) This function measured fitness as a function of the intra cluster distance. The the personal best and the global best were then updated if the function returned an improved value. The particle velocity, v_i, and the centroids, x_i, updated using the formulas,

$$v_{i,k}(t+1) = wv_{i,k}(t) + c_1U(0,1)(y_{i,k}(t) - x_{i,k}(t)) + c_2U(0,1)(\hat{y}_{i,k}(t) - x_{i,k}(t))$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

(Engelbrecht, 2007) The values of c_1 and c_2 were user tunable parameters which determined the ratio of influence that the global best and the personal best had on movement of the particle. Over a series of tests it was determined that the best results were found by exploiting the global best and only using the personal best to anchor the particles.

The inertia of the particle was represented by w, which adjusted the continued movement of the particle in the direction of the previous velocity.

$$w = N(0.72, 1.2)$$

(Engelbrecht, 2007) Velocity clamping was implemented to prevent the particles from moving too quickly and passing over possible solutions. The velocity of each centroid for each dimension was limited using,

$$v_{i,j}(t+1) = \begin{cases} v'_{i,j}(t+1) & : v'_{i,j}(t+1) < V_{max,j} \\ V_{max,j} & : v'_{i,j}(t+1) \geq V_{max,j} \end{cases}$$

$$v_{max,j}(t+1) = \begin{cases} \gamma V_{max,j}(t) & : f(\hat{y}(t)) \geq f(\hat{y}(t-t'))\forall t' = 1, ..., n_t \\ V_{max,j} & : otherwise \end{cases}$$

(Engelbrecht, 2007) Where the maximum velocity is decreased if there has not been a change in the global best after 10 generations. The value of γ was linearly adjusted so that it decreased from 1 to 0.01. This decreased the velocity faster at the end of the runtime to speed the fine tuning of the solutions as the solutions were allowed less movement.

Dataset	Number of Particles	c_1	c_2	Initial V_{max}
Breast Cancer	10	.06	.7	.3
Cardiotocography	15	.1	.7	.3
Climate Model	10	.1	.7	.4
E. Coli	30	.1	.7	.4
Glass	30	.1	.7	.4
Hill Valley	10	.1	.7	.4
Liver	10	.1	.7	.4
Ionosphere	15	.1	.8	.4
Knowlege	10	.1	.7	.3
Fertility	10	.1	.7	.3

4.4 Cluster Metrics

Because the goals of clustering are to minimize intra-cluster distance and maximize inter-cluster distance, we wanted a metric that could evaluate both at the same time. The metric we used for our evaluation was the Davies-Bouldin index, which is given by

$$DB = \frac{1}{n}\sum_{i=1}^{n} max_{i\neq j}(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)})$$

where σ_x is the average distance from each point in cluster x to the center of the cluster, and $d(c_i, c_j)$ is the distance between the centers of clusters i and j. Since we want our clusters to be dense and very separated, a lower Davies-Bouldin index is preferable.

5. Results

Each dataset was run one hundred times on each of the ten data sets. Each run the algorithm was given 300 iterations to cluster the data. The results of each run were averaged, creating a value that was representative of the algorithms performance.

The results show that the on average the competitive algorithm performed the worst, with the least inter-cluster distance and due to the relatively high intra-cluster distance it performed poorly on the Davies-Boudin index. This is in contrast to the K-Means algorithm which on average had a higher inter-cluster distance and a similar inter-cluster distance and so scored the lowest on the Davies-Boudin. The ant colony performed comparably to the K-means algorithm despite averaging slightly lower intra and inter cluster distances. The particle swarm algorithm performed poorly on intra cluster distance, and the inter-cluster distance was not high enough to compensate, leaving it behind both the ant colony and the K-means algorithm.

6. Discussion

The results of the experiments show the value of both centroid based clustering as it is used by the Particle Swarm algorithm and K-Means, and hierarchical clustering as it is used by the ant colony.

Where the advantages of Ant Colony Optimization, and Particle Swarm algorithm are seen is in consistency. The two algorithms achieved much less variation across different

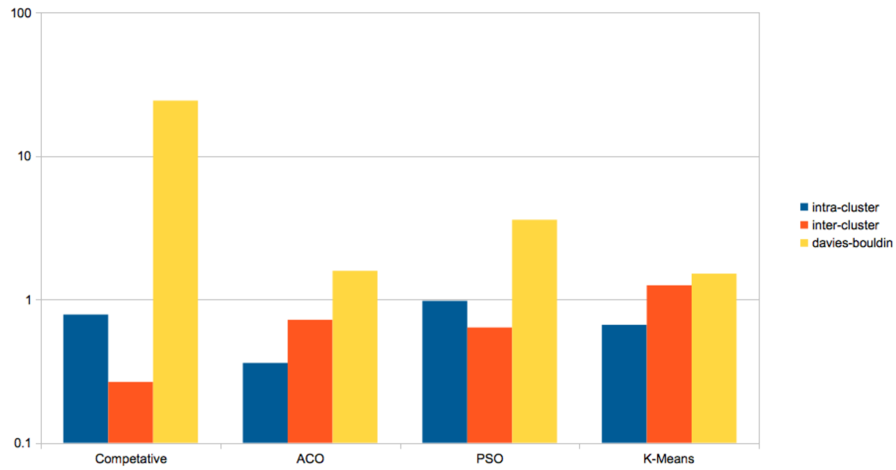

Figure 1: The average over all the data sets for each of the algorithms

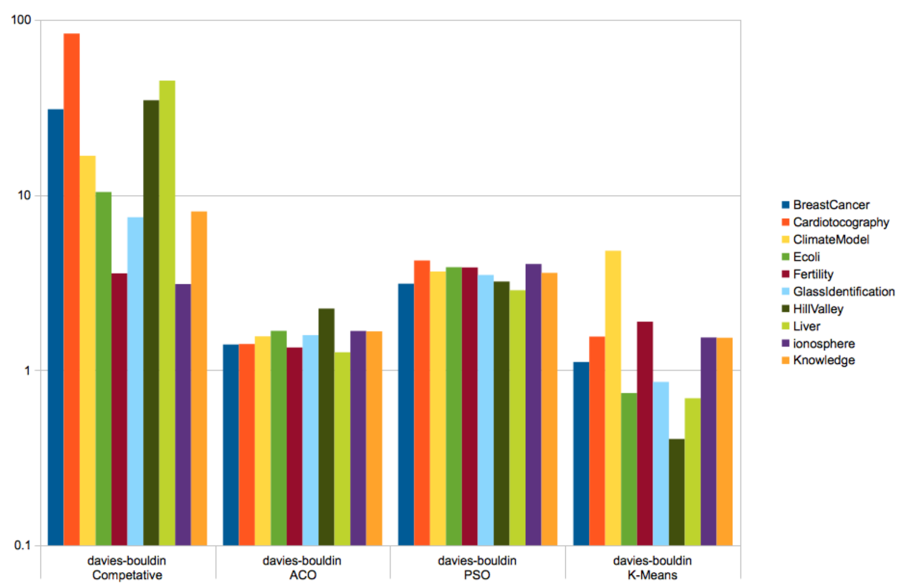

Figure 2: The average results of the Davies-Bouldin index for the different algorithms for each of the different data sets

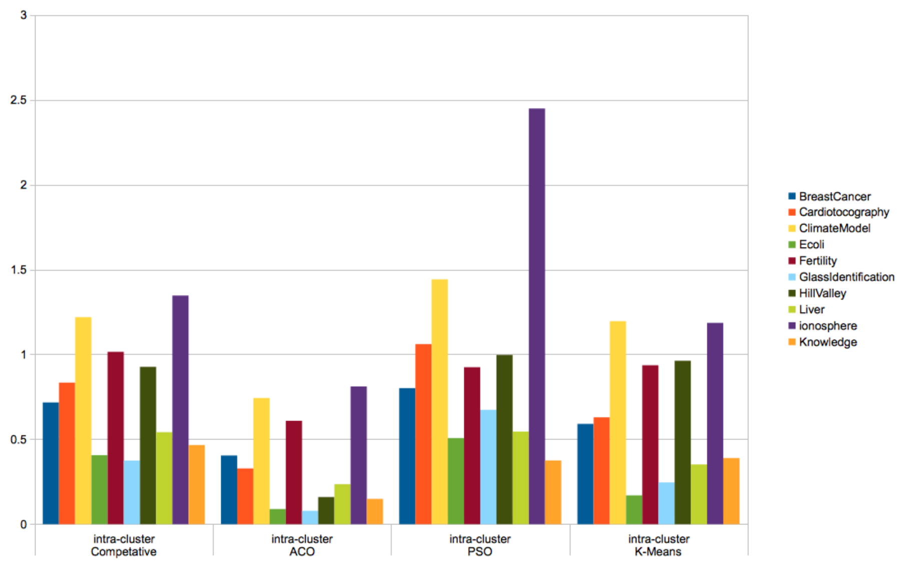

Figure 3: The average intra-cluster distance for the different algorithms for each of the different data sets

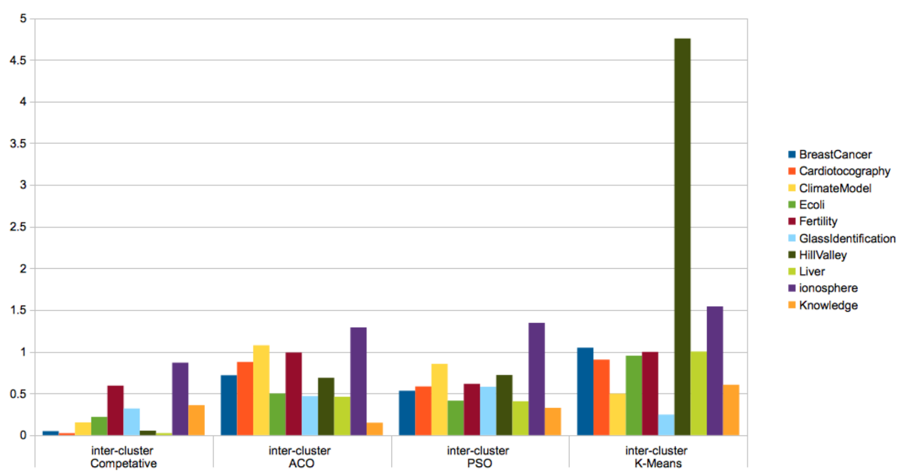

Figure 4: The average inter-cluster distance for the different algorithms for each of the different data sets

data sets. This indicates that it is less sensitive to the characteristics of the data, whereas the performance of the K-Means, and competitive learning algorithms change significantly based on the input dataset.

It is clear that the competitive neural network seemed to perform the least effectively of the four methods discussed in this paper. It seemed to create fairly dense clusters, but where its fitness was shown to be exceptionally weak was with maximizing inter-cluster distance. This could possibly have been caused by a large portion of the data set being classified as one cluster.

Another important consideration to take into account is the effect that the hierarchical clustering algorithm that was used to extract the clusters from the Ant Colony. Any outlying points, or points that didnt have time to reach their cluster may have been misplaced on the map when the clusters were extracted. This could have skewed the intra and extra cluster distances as the point could have been included in a cluster to which it should not have belonged.

The performance of the K-Means algorithm shows the potential of this algorithm, and justifies its use as a reference clustering algorithm. Improved clusterings could be made possible by combining this algorithm with the K-Means algorithm(Van der Merwe, 2003). This would allow for the benefits of both algorithms to be combined by using the K-Means clusters to seed the initial centroids used in the PSO.

Despite falling behind the K-Means clustering algorithm, the biologically inspired algorithms have shown to be adept at unsupervised clustering tasks. However the inconsistent behavior of the competitive learning neural network indicates that additional research and refinement of the algorithm may be required before it become a useful tool for unsupervised clustering tasks.

7. Conclusion

These experiments showed that the performance of the swarm based algorithms were at least comparable to traditional clustering algorithms. In general, they both seemed to regularly out perform the competitive neural network and had results very similar to k-means.

References

Eberhart, R., & SHI, Y. (2007). Computational intelligence. Burlington MA: Morgan Kaufman.

Handl, J., Knowles, J. D., & Dorigo, M. (2003, December). On the Performance of Ant-based Clustering. In HIS (pp. 204-213).

Van der Merwe, D. W., & Engelbrecht, A. P. (2003, December). Data clustering using particle swarm optimization. In Evolutionary Computation, 2003. CEC'03. The 2003 Congress on (Vol. 1, pp. 215-220). IEEE.

Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.