Design Document: Comparing Clustering/Unsupervised Learning Algorithms

Amy Peerlinck, Taylor Heinecke, Na'Shea Wiesner, Scott Martin

November 27, 2017

1 Description of the Project

Clustering is a data mining task with the goal of grouping similar data together. Five different clustering methods will be implemented and compared in this project: two traditional clustering methods, DB-Scan and K-Means clustering, a competitive learning neural network (CL) as an alternative to the traditional methods, and finally particle swarm optimization (PSO) and ant colony optimization (ACO).

K-means clustering implements a matrix containing the data points and dimensions, and a matrix containing the initial cluster centers and the dimensions. Using the distance between data points and the cluster center, k-means looks for the locally optimal cluster sum for that data point and each cluster [4].

DBSCAN stands for Density Based Spatial Clustering of Applications with Noise, and finds all points that are "density" reachable from a certain point p. If p is a center point, it forms a cluster, if it is an edge point there are no other points "density" reachable, and it moves on to the next point [3].

A neural network implementing competitive learning, starts with weighted vectors which are updated based on the distance between a randomly chosen point and the closest cluster center (also called the winner). This results in the closest centroid having the highest weight [8].

ACO is inspired by the behavior of real ants, where an ant is defined to be a computational agent. Each ant iteratively constructs a solution for the problem and moves from state to state based on a larger partial solution. At each step, an ant computes a set of possible moves from its current state, and moves according to a probability distribution based on the desirability of the move and how profitable it was previously. This eventually results in the optimal path [6].

PSO is based on how individuals interact with each other. Particles are placed in the search space of a problem and they evaluate their position. Each particle then determines their next move by combining their history of its current and historically best locations with those of one or more members of the swarm.

Once all particles have been moved for a number of iterations, the swarm as a whole is supposed to find a result close to the optimal solution [7].

Both ACO and PSO are altered for clustering use, and these alterations are explained in Section 3. All the aforementioned clustering algorithms will be performing partition- or density-based clustering, the hierarchical clustering problem will not be considered. The goal of the project is to compare these five algorithms on five different data sets chosen from the UCI ML library [5], where each algorithm will be tested on each data set. This is explained in more detail in the Experimental Design (Section 4).

2 Software Architecture

The architecture for the different algorithms is detailed in Figure 1, also including the data processing. Each algorithm is its own class and is part of the *Clustering* package. The *Clustering* package is then fully imported into the *Init* class where the algorithms are run. Furthermore, the *Init* class receives the transformed data, which is the original data that has been been altered to fit the clustering algorithms. Because all the clustering algorithms need to calculate a distance, there is a separate *Distance* package that currently only contains the Euclidean distance metric as a class. The PSO and ACO algorithms implement the Particle and the Grid and Ant classes respectively.

3 Design Decisions

Several software engineering techniques were considered when planning out the architecture. The most notable concepts will be explained briefly. As mentioned earlier, a separate package was implemented for the distance metrics, even though there is only one metric used by all five algorithms. This decision was made in order to make it easier to implement other means of measuring distance, in case this would prove to be necessary. Furthermore, the composition classes used in the PSO and ACO algorithms (Particle, Ant and Grid) were used to prevent the existence of "God" classes, by separating some of the functionality. The functionality for each of these algorithms is explained in more detail in the following sections. K-Means, DB-Scan, and the CL Neural Network were all implemented in the conventional way, with no major implementation decisions made.

3.1 Ant Colony Optimization

For ACO clustering, it was decided to use a grid implementation that contains the probability of a cluster for certain data based on pheromones. This is called a pheromone matrix, and is randomly initialized. After the ants have traversed the grid, the cluster with the highest level of pheromones is chosen for the data.

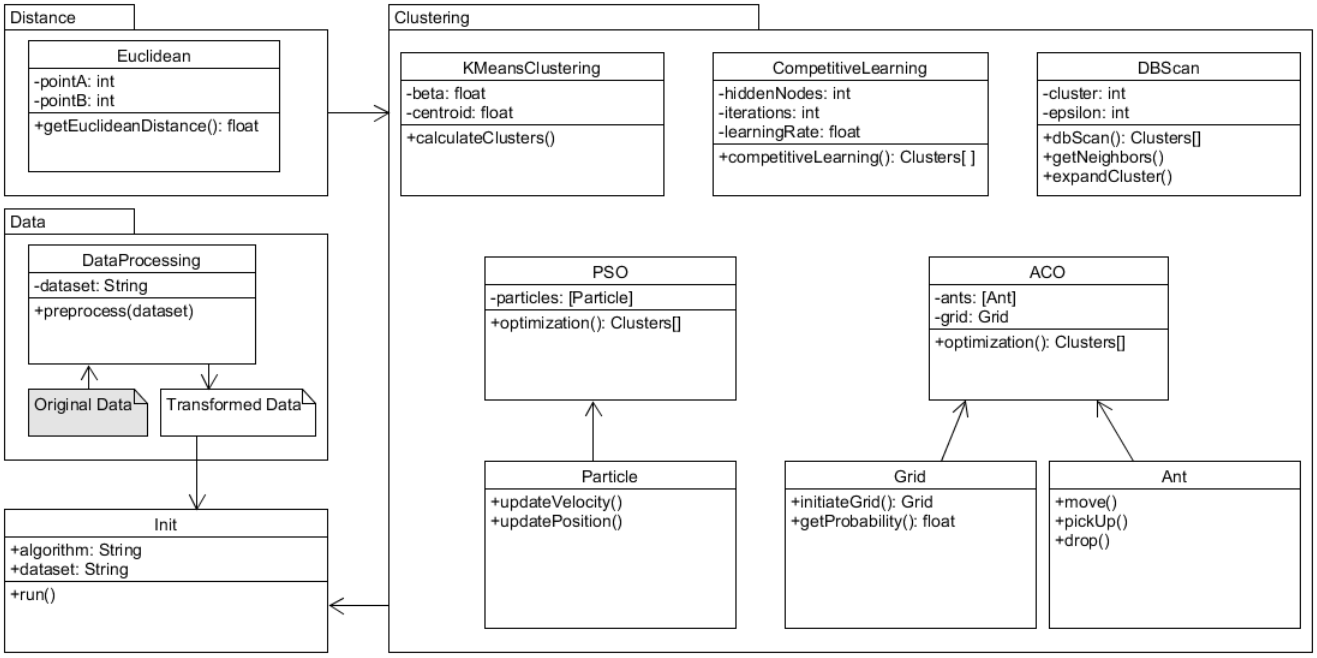

Figure 1: UML Diagram of software architecture

3.2 Particle Swarm Optimization

For the PSO clustering implementation, a Gauss chaotic map will be used to avoid being trapped in local optima. This allows for a total analysis of quantitative and qualitative properties of chaos [2]. This replaces *phi1* and *phi2* from the update equation in the original algorithm, and instead, uses the Gauss chaotic map to update the velocity of a particle in the following way:

$$v_i' = wv_i + Gr_1(pbest_i - x_i) + Gr_2(gbest - x_i) \tag{1}$$

where w is the inertia factor preventing the velocity value from escaping the bounds of the search space. From the Gauss chaotic map, 2 random values between 0 and 1 are extracted and represented by Gr_1 and Gr_2. These values are updated by $\frac{1}{x} mod 1$ if the number is between 0 and 1, and remains 0 otherwise. The position of a particle is updated by adding the previous position to the newly calculated velocity v_i'.

4 Experimental Design

In order to draw significant conclusions from the experiments, and see how the different algorithms perform under different circumstances, five data sets are

implemented in the project. Abalone, epileptic, contraceptive method choice, US census and water treatment data. Each of the algorithms is run on these five data sets and are compared based on the cohesion, separation, and number of clusters formed from their results.

4.1 Hypothesis

K-means clustering is generally used for high-dimensionality data and is thus expected to perform best on data sets with more attributes. More specifically, it is thought to create a relatively high number of high cohesion clusters. DB-SCAN on the other hand is thought to have less clusters and thus will probably have lower cohesion but higher separation, at a faster convergence rate than K-means. Because the CL NN has a parameter defining the maximum amount of clusters, it can be manipulated to have as many clusters as desired. However, the max number of clusters does not have to be reached, so it is expected to have a lower amount of clusters than K-means, but more than DB-SCAN. Because ACO is based on the behaviour of ants, it does not move dissimilar points away from each other, resulting in poor separation, but high cohesion. PSO is expected to have higher separation but less cohesion than ACO. Both ACO and PSO should converge proportionate to the amount of data points in the data set to be clustered.

4.2 Evaluation Metrics

To compare the different clustering algorithms, two different measurements are calculated, cohesion and separation. Cohesion calculates how closely knit the clusters themselves are, also known as intra-variance. This is measured by calculating the average distance between the data points and the cluster's centroid. Separation checks how separate the clusters are from each other by calculating the weighted average distance between cluster centroids. This is also known as inter-variance [1]. Furthermore, convergence rate and number of clusters formed will be compared for each of the algorithms.

References

[1] Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M. Pérez, and Iñigo Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243 – 256, 2013.

[2] Li-Yeh Chuang, Yu-Da Lin, Hsueh-Wei Chang, and Cheng-Hong Yang. Snp-snp interaction using gauss chaotic map particle swarm optimization to detect susceptibility to breast cancer. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pages 2548–2554. IEEE, 2014.

[3] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters a density-based algorithm for dis-

covering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

[4] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[5] M. Lichman. UCI machine learning repository, 2013.

[6] Vittorio Maniezzo and Antonella Carbonaro. *Ant Colony Optimization: An Overview*, pages 469–492. Springer US, Boston, MA, 2002.

[7] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. *Swarm Intelligence*, 1(1):33–57, Jun 2007.

[8] L. Xu, A. Krzyzak, and E. Oja. Rival penalized competitive learning for clustering analysis, rbf net, and curve detection. *Trans. Neur. Netw.*, 4(4):636–649, July 1993.