

Hard c-means Clustering

The classical ‘hard’ c-means clustering algorithm has the following characteristics,

- Each data point is a member of ONE and only ONE cluster;
- The number of clusters, c , necessary to correctly cluster the data is known *a priori*.

Moreover,

$$2 \leq c < P$$

where, P is the number of data points.

More formally we need to identify the characteristic function, χ , relating each data point, x_k , to one of a family of sets $\{A_i, i = 1, \dots, c\}$

Thus,

$$\chi_{A_i}(x_k) \begin{cases} 1 & \text{if } x_k \in A_i \\ 0 & \text{if } x_k \notin A_i \end{cases}$$

Some properties of the (hard c-means) clustering process

Property 1 – the union of all cluster (sets), A_i , spans the set of data points, X .

$$\bigcup_{i=1}^c \chi_{A_i}(x_k) = 1 \quad \forall k$$

Property 2 – there is no overlap between clusters.

$$\chi_{A_i}(x_k) \cap \chi_{A_j}(x_k) = 0 \quad \forall k$$

Property 3 – clusters cannot be empty, and cannot contain all data points.

$$0 < \sum_{k=1}^P \chi_{A_i}(x_k) < P \quad \forall i$$

Defining a Cluster

So far we have just set the scene – nothing has been said about how data points are related to a specific cluster.

Let matrix, U , be a $c \times P$ matrix of assignments between data points and clusters.

That is to say, if $\chi_{ij} \equiv \chi_{A_i}(x_j)$ represents the membership (0 or 1) between the j th data point and i th cluster, then,

- U is a matrix of χ_{ij} ($i = 1, \dots, c; j = 1, \dots, P$)

Let M_{cP} be the universe of hard ‘ c ’ partitions, or

- The allocation of memberships (0 or 1) such that each data point is associated with one class;

$$M_{cP} = \left\{ U \mid \chi_{ij} \in \{0,1\}; \sum_{i=1}^c \chi_{ik} = 1; 0 < \sum_{j=1}^P \chi_{ik} < P \right\}$$

Ranking Clusters

- What represents a good cluster and what represents a bad cluster?
- *Objective function* differentiates between *quality* of different cluster allocations.
- *C-means* algorithm uses a sum of distances between,
 - Proposed cluster and,
 - Associated data points belonging to this cluster.
- Objective is to find the best centroid and allocation of data points such that the distance is minimized.

That is we wish to minimize,

$$J(U, V) = \sum_{k=1}^P \sum_{i=1}^c \chi_{ik} (d_{ik})^2$$

where, d_{ik} is a suitable distance metric, say an Euclidean norm, between the k th data sample, x_k , and i th cluster center v_i ,

$$d_{ik} = d(x_k - v_i) = \|x_k - v_i\| = \left[\sum_{j=1}^m (x_{kj} - v_{ij})^2 \right]^{\frac{1}{2}}$$

i.e. each data point lies in an ‘ m ’ dimensional space.

We need the optimal combination, (U^*, V^*) , minimizing $J(U, V)$, or

$$J(U^*, V^*) = \min_{U \in M_{cP}} \{J(U, V)\}$$

Why is this problem difficult?

- Number of possible partition matrices increases very quickly with P and c ;
- Relation between U and V additionally requires optimization.

GA approach to hard c-means clustering

Design assumption,

- Concentrate on finding the cluster center location, \underline{v}_i .
- Partitioning matrix, U , merely follows a nearest neighbour allocation with respect to the cluster centers.

Objective now in terms of v alone or,

Minimize,

$$R_1(v) = \sum_{k=1}^P \min(d_{1k}, d_{2k}, \dots, d_{ck})$$

Individuals

- Representation
 - Each individual needs to represent a set of cluster centers,
 - Consider case of a binary vector,
 - How many bits are required?
 - Use a gray code – any two adjacent bits are 1 bit different.
- Search Operators
 - Does crossover face any constraints?
- What is the fitness function?

Overall GA Algorithm

1. Choose $\{m, c, b, d_{ik}, P(\text{crossover}), P(\text{mutation})\}$;
2. Initialize population, size N , such that points equally cover each dimension, m , of the cluster space;
3. Calculate $R_1(v)$ for each member of the population
4. Convert population members to equivalent binary gray code;
5. For $g = 1$ to max generations;
 - a. Select $N/2$ parent pairs;
 - b. Stochastically apply 2 point crossover;
 - c. Calculate $R_1(v)$ for each child;
 - d. Compose new population from best children and parents;

6. Return individual with minimal $R_I(v)$.

Notes:

- $P(\textit{mutation}) = 1.75 / (N \times (\textit{bits per individual})^{0.5})$
- Modified objective function: $R_I'(v) = (1 + \alpha) R_I(v)$
 - where $\alpha \in [0, c]$
- $P(\textit{crossover}) = 0.9$

Evaluation – 6 Datasets

- Synthetic 3-D data
 - $y = (1 + x_1^{-2} + x_2^{-1.5})^2$
 - 50 data points over the interval $\{1 \leq x_1, x_2 \leq 5\}$
 - Exhaustive enumeration indicates that 6 clusters are expected.
 - 20,000 trials (initializations) using classical approach
 - 50 populations in GA
- Real world data
 - Iris benchmark
 - 2 features;
 - 3 classes of Iris, 50 patterns per class.
 - Objective
 - allocation of clusters to correspond to known pattern labels
 - 9,000 trials using classical approach
 - 50 populations in GA
 - Multiple Sclerosis
 - 5 features;
 - 2 classes – 29 patterns represent MS; 69 without MS.
 - Objective
 - allocation of clusters to correspond to known pattern labels
 - 9,700 trials using classical approach
 - 50 populations in GA

Classical Hard c-means with Euclidean Norm for Synthetic function, MS and Iris problems

Partition	Synthetic function		MS		Iris	
	Cost	Count	Cost	Count	Cost	Count
1	0.935	3781	82494.57	6802	78.941	2857
2	1.189	1867	82527.961	2259	78.945	3929
3	2.181	1484	162345.281	10	142.852	18
4	2.304	1641	178135.359	629	142.859	151
5					142.879	34
6					142.218	1
7					143.454	307
8					145.279	70
9					152.369	1049
10					680.824	584

Genetic Hard c-means over 50 trials with Euclidean Norm for Synthetic feature, MS and Iris problems

Data Set	Population size	Generations	Mutation	Average Values		Lowest Cost
				Cost	SD	
Synthetic function	30	550	0.046	0.947	0.05	0.935
	30	550	0.03	0.951	0.06	0.935
	50	550	0.02	0.945	0.05	0.935
	50	550	0.03	0.935	0.01	0.935
MS	50	700	0.0113	82499.281	11.584	82494.57
	50	800	0.0088	82499.578	10.843	82494.57
	50	3300	0.0088	82499.578	0.003	82494.57
	75	800	0.0088	82499.250	4.674	82494.57
Iris	30	750	0.0015	78.843	0.002	78.941
	30	750	0.003	78.843	0.002	78.941
	50	550	0.0015	78.843	0.002	78.941
	50	550	0.0015	78.842	0.002	78.941
	75	400	0.0064	78.842	0.002	78.941

Notes

- Degenerate solutions reduce the effectiveness of the classical approach
 - 58% of initializations on Synthetic function are not optimal;
 - 68% of initializations on Iris not optimal;
 - 70% of initializations on MS are not optimal;
- GA appears to be robust to changes in parameters, but
 - Computational overhead in operation
 - However, parallel implementation is straightforward

Reference

- Hall L.O., Ozyurt I.B., Bezdek J.C. (1999) IEEE Transactions on Evolutionary Computation 3(2), pp 103-112
 - Additional results on image segmentation (quantization);

Additional Reading

- M. Halkidi, Y. Batistakis, M. Vazirgiannis (2001) Clustering algorithms and validity measures”. Tutorial paper, *Proceedings of SSDBM Conference*, pp 3-22.
 - On determining the quality of your clusters!
- Julia Handl and Joshua Knowles (2007) An Evolutionary Approach to Multiobjective Clustering. IEEE Transactions on Evolutionary Computation 11(1): 56-76.
 - Pareto multi-objective methods provide the opportunity to define the optimization goal from multiple perspectives. As such they have an implicit advantage over single objective methods. Delivering on this goal has been somewhat elusive... up to now☺
- Y. Kim, W. Street and F. Menczer (2000) Feature Selection for Unsupervised Learning via Evolutionary Search. Proceedings of ACM International Conference on Knowledge Discovery and Data Mining (KDD). Pp 365 – 369.
 - Subspace clustering identifies a common subset of attributes for building the clusters. GA used to perform the attribute selection and ‘ k ’ in k -NN clustering. k -NN clustering algorithm appears in the inner loop to identify the clusters.