Appendix A – K-means and Fuzzy K-means Clustering

11.1 K-Means Clustering

Clustering algorithms attempts to segregate a dataset into distinct regions of membership, this is widely performed by a gradient descent based iterative algorithm that is known as k-means (or c-means) algorithm or the Generalised Lloyd algorithm {Karayiannis 95]. The k-means algorithm begins with a set of *k* initial exemplars, where the data is to be segregated into *k* distinct regions. Each region is evaluated with the exemplar as the centeroid of the region. Data points are assigned to the exemplar in a nearest neighbour fashion and the exemplars moved to minimise the distance between the exemplar and its members. This membership is reassessed at each iteration and repeated until the algorithm converges upon a solution i.e. the movement of the exemplars approaches zero.

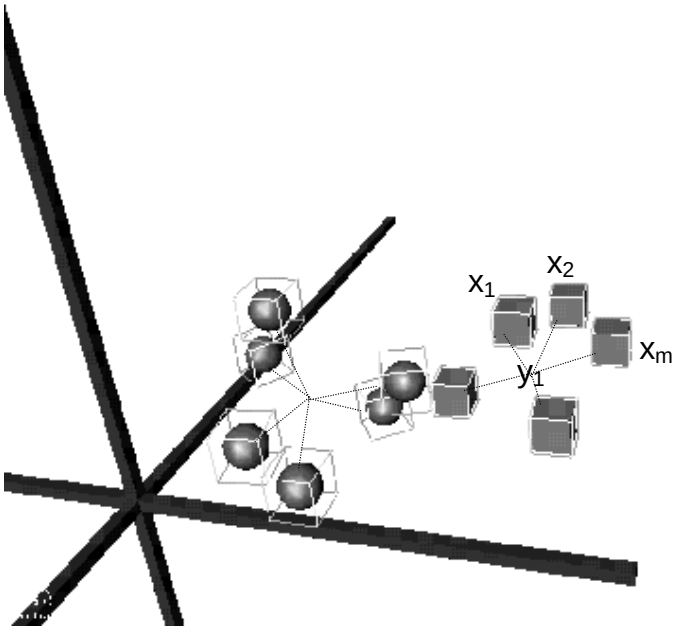

Figure 11.1.1 - K-means clustering

For the clustering of a training set $X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M)$ where $\mathbf{x}_i \in \Re^n$ is an *n* dimensional vector in Euclidean space and $i = 1,2,\ldots,M$. The segregation of the training set into *k* clusters using the exemplars (cluster centres)

$Y = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M)$ where $\mathbf{y}_j \in \Re^n$ and $j = 1,2,\ldots,k$ is performed by minimising the cost function *D* where,

$$D = \frac{1}{M}\sum_{i=1}^{M} d_{\min}(\mathbf{x}_i) = \frac{1}{M}\sum_{i=1}^{M} \min_{\mathbf{y}_j \in y}\left(d(\mathbf{x}_i, \mathbf{y}_j)\right)$$

The K-means algorithm assigns each training vector to a certain cluster on the basis of the nearest neighbour condition. According to this strategy, the training vector $\mathbf{x}_i$ is assigned to the j^{th} cluster if $d(\mathbf{x}_i, \mathbf{y}_j) = d_{\min}(\mathbf{x}_i) = \min_{\mathbf{y}_j \in Y} d(\mathbf{x}_i, \mathbf{y}_j)$, where $d(\mathbf{x}_i, \mathbf{y}_j)$ is the squared Euclidean distance between the training vector $\mathbf{x}_i$ and the exemplar $\mathbf{y}_j$, defined as $d(\mathbf{x}_i, \mathbf{y}_j) = \left\|\mathbf{x}_i, \mathbf{y}_j\right\|^2$ [Karayiannis 95].

The nearest neighbour description can be described by the membership function *u*,

$$u_j(\mathbf{x}_i) = \begin{cases} 1 & \text{if} \quad d(\mathbf{x}_i, \mathbf{y}_j) = d_{\min}(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases}$$

The algorithm minimises this cost function D through the iterative refinement of cluster centres where the exemplar $\mathbf{y}_j$ is the mean of the vectors assigned to it,

$$\mathbf{y}_j = \frac{\sum_{i=1}^{M} u_j(\mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{M} u_j(\mathbf{x}_i)} \text{ and } j = 1,2,\ldots,k$$

Although the k-means algorithm is simple and relatively fast to iterate it is a gradient descent method and therefore only capable of finding local energy minima. It will always converge on a low cost solution, but because the energy surface that it traverses is full of local minima, it will not necessarily find the global solution As such, it is extremely sensitive to the initial placement of exemplars. Exemplars are commonly placed randomly within the data space or randomly allocated from the data points themselves. It is therefore necessary to

run the algorithm a number of times with different random initialisations to try and find the best local minima possible.

11.2 Selecting the Natural Number of Clusters k

Often during clustering the natural number of distinct clusters is known. Under these circumstances cluster analysis can be performed using *k*=5. However, more often, little is known about the nature of the data and a method of estimating *k* is required. Furthermore, the nature of the energy minimisation within the k-means algorithm makes the assumption that clusters are hyper-spherical. Where elongated hyper-elliptical clusters are present these may be better modelled using multiple adjoining spherical clusters as demonstrated in chapter 5.2.

The cost function D is commonly used as a metric with which to assess the performance of clustering. As the number of clusters is increased, so the resulting overall cost diminishes in a characteristic way.

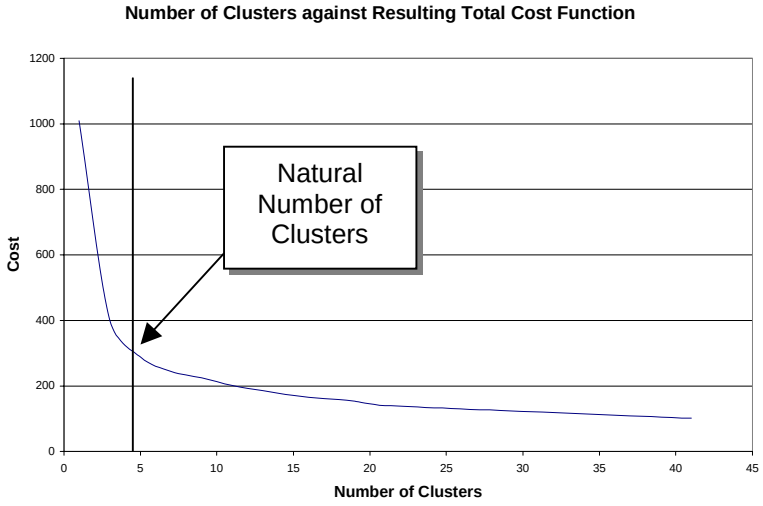

Figure 11.2.1 - Characteristic Cost Graph for k-means for 1 < k < M

Figure 11.2.1 shows the characteristic graph produced for a training set by plotting the resulting overall cost of a converged solution against the number of clusters *k*, where $1<k<M$. The overall cost of a solution decreases as the number of clusters increases, where $k=1$ produces the highest cost and $k=M$ (the number

of training examples) produces a cost of zero. However, as the number of *k* is increased there becomes a point where increasing *k* further does not produce a significant decrease in the resulting cost. This is said to be the natural number of clusters of the data and is a simple but effective method for estimating *k*.

11.3 The Fuzzy K-means Algorithm (FCM)

Fuzzy set theory is a method of representing vagueness in every day life. Bezdeck, Ehrlich and Full proposed a family of fuzzy k-means algorithms [Bezdeck 84]. Fuzzy clustering algorithms consider each cluster as a fuzzy set, while a membership function measures the possibility that each training vector belongs to a cluster. As a result, each training vector may be assigned to multiple clusters with some degree of certainty measured by the membership function. Thus, the partition of the training set is based upon soft decisions [Karayiannis 95].

The fuzzy k-means algorithm uses a fuzzy membership rule where [Bezdeck84]

$$u_j(\mathbf{x}_i) = \frac{1}{\sum_{\ell=1}^{k} \left(\frac{d(\mathbf{x}_i, \mathbf{y}_j)}{d(\mathbf{x}_i, \mathbf{y}_\ell)} \right)^{\frac{1}{m-1}}}$$

The new cluster position $\mathbf{y}_j$ is therefore calculated as

$$\mathbf{y}_j = \frac{\sum_{i=1}^{M} u_j(\mathbf{x}_i)^m \mathbf{x}_i}{\sum_{i=1}^{M} u_j(\mathbf{x}_i)^m} \text{ and } j = 1,2,\ldots,k$$

The "fuzziness" of the clustering produced by these algorithms is controlled by the parameter *m*, which is greater than 1 [Bezdeck84]. As this parameter approaches 1, the partition of the data is nearly the binary decision used in the k-means algorithm. However, as the parameter *m* is increased the membership degrades towards a fuzzy state [Bezdeck84].

Results comparing the partition of space using the k-means algorithm and the FCM algorithm can be found in section 5.2, Figure 5.4.3.