Singapore University of Technology and Design
Information Systems Technology and Design
50.007 Machine Learning
Clustering: K-Medoids

K-medoids is a slight variant of the K-means algorithm.

1 The K-medoids algorithm

We had previously defined the cost function for the K-means algorithm in terms of squared Euclidean distance of each point $x^{(i)}$ to the closest cluster representative. We showed that, for any given cluster, the best representative to choose is the mean of the points in the cluster. The resulting cluster mean typically does not correspond to any point in the original dataset. The K-medoids algorithm operates exactly like K-means but, instead of choosing the cluster mean as a representative, it chooses one of the original points as a representative, now called an *exemplar*. Selecting exemplars rather than cluster means as representatives can be important in applications. Take, for example, Google News, where a single article is used to represent a news cluster. Blending articles together to evaluate the "mean" would not make sense in this context. Another advantage of K-medoids is that we can easily use other distance measures, other than the squared Euclidian distance.

The K-medoids objective is very similar to the K-means objective:

$$Cost(C^1, \ldots, C^k, z^{(1)}, \ldots, z^{(k)}) = \sum_{j=1}^{k} \sum_{i \in C^j} d(x^{(i)}, z^{(j)}) \tag{1}$$

The algorithm:

1. Initialize exemplars: $\{z^{(1)}, \ldots, z^{(k)}\} \subseteq \{x^{(1)}, \ldots, x^{(n)}\}$ (exemplars are k points from the original dataset)
2. Repeat until there is no further change in cost:
   (a) for each j: $C^j = \{i : x^{(i)}\text{'s closest exemplar is } z^{(j)}\}$
   (b) for each j: set $z^{(j)}$ to be the point in C^j that minimizes $\sum_{i \in C^j} d(x^{(i)}, z^{(j)})$

In order to update $z^{(j)}$ in step (b), we can consider each point in turn as a candidate exemplar and compute the associated cost. Among the candidate exemplars, the point that produces the minimum cost is chosen as the exemplar.