CLUSTERING ALGORITHMS VIA FUNCTION OPTIMIZATION

- In this context the clusters are assumed to be described by a parametric specific model whose parameters are unknown (all parameters are included in a vector denoted by $\underline{\theta}$).

  Examples:

  - Compact clusters. Each cluster C_i is represented by a point $\underline{m}_i$ in the l-dimensional space. Thus $\underline{\theta}=[\underline{m}_1^T, \underline{m}_2^T, \ldots, \underline{m}_{\mathrm{m}}^T]^T$.
  - Ring-shaped clusters. Each cluster C_i is modeled by a hypersphere $C(\underline{c}_i, r_i)$, where $\underline{c}_i$ and r_i are its center and its radius, respectively. Thus

$$\underline{\theta}=[\underline{c}_1^T, r_1, \underline{c}_2^T, r_2, \ldots, \underline{c}_m^T, r_m]^T.$$

- A cost $J(\underline{\theta})$ is defined as a function of the data vectors in X and $\underline{\theta}$. Optimization of $J(\underline{\theta})$ with respect to $\underline{\theta}$ results in $\underline{\theta}$ that characterizes optimally the clusters underlying X.
- The number of clusters m is a priori known in most of the cases.

❖ Hard Clustering Algorithms:

Each vector belongs exclusively to a single cluster. This implies that:

- $u_{ij} \in \{0, 1\}, \quad j=1,\ldots,m$
- $\sum_{j=1}^{m} u_{ij} = 1$

That is, it can be seen as an extreme special case of the fuzzy algorithmic schemes.

However, now, the cost function

$$J(\underline{\theta}, U) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij} d(\underline{x}_i, \underline{\theta}_j)$$

is not differentiable with respect to $\underline{\theta}_j$.

Despite that, the two-step optimization procedure (with respect to u_{ij}'s and with respect to $\underline{\theta}_j$'s) can be applied, taking into account that, for fixed $\underline{\theta}_j$'s, the u_{ij}'s that minimize $J(\underline{\theta}, U)$ are chosen as

$$u_{ij} = \begin{cases} 1, & if \ d(\underline{x}_i, \underline{\theta}_j) = \min_{k=1,\ldots,m} d(\underline{x}_i, \underline{\theta}_k) \\ 0, & otherwise \end{cases}, \quad i = 1,\ldots,N$$

❖ Hard Clustering Algorithms (cont.)

➢ *Generalized Hard Algorithmic Scheme (GHAS)*

- Choose $\underline{\theta}_j(0)$ as initial estimates for $\underline{\theta}_j$, $j=1,\ldots,m$.
- $t=0$
- Repeat
  - For $i=1$ to N
    - For $j=1$ to m

Determination of the partition:

$$u_{ij}(t)=\begin{cases}1, & if\ \ d(\underline{x}_i,\underline{\theta}_j(t))=\min_{k=1,\ldots,m} d(\underline{x}_i,\underline{\theta}_k(t))\\ 0, & otherwise\end{cases}, \quad i=1,\ldots,N$$

    - End {For-j}
  - End {For-i}
  - $t=t+1$

❖ Hard Clustering Algorithms (cont.)

➢ *Generalized Hard Algorithmic Scheme (GHAS) (cont.)*

– For j=1 to m

o *Parameter updating:* Solve

$$\sum_{i=1}^{N} u_{ij}(t-1)\frac{\partial d(\underline{x}_i, \underline{\theta}_j)}{\partial \underline{\theta}_j} = 0$$

o with respect to $\underline{\theta}_j$ and set $\underline{\theta}_j(t)$ equal to the computed solution

– End {For-j}

- Until a termination criterion is met

➢ Remarks:

- In the update of each $\underline{\theta}_j$, only the vectors $\underline{x}_i$ for which $u_{ij}(t\text{-}1)=1$ are used.
- GHAS may terminate when either
  - $\|\underline{\theta}(t)\text{-}\underline{\theta}(t\text{-}1)\|<\varepsilon$ or
  - U remains unchanged for two successive iterations.

❖ Hard Clustering Algorithms (cont.)

➢ The K-Means Algorithm

General comments

- It is a special case of GHAS where
  - Point representatives are used.
  - The squared Euclidean distance is employed.
- The cost function $J(\underline{\theta}, U)$ becomes now

$$J(\underline{\theta}, U) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij} \, \| \underline{x}_i - \underline{\theta}_j \|^2$$

❖ Hard Clustering Algorithms (cont)

➢ The k-Means algorithm

- Choose arbitrary initial estimates $\underline{\theta}_j(0)$ for the $\underline{\theta}_j$′ s, $j=1,\ldots,m$.
- Repeat
  - For $i=1$ to N
    - Determine the closest representative, say $\underline{\theta}_j$, for $\underline{x}_i$
    - Set $b(i)=j$.
  - End {For}
  - For $j=1$ to m
    - *Parameter updating:* Determine $\underline{\theta}_j$ as the mean of the vectors $\underline{x}_i \in X$ with $b(i)=j$.
  - End {For}
- Until no change in $\underline{\theta}_j$′ s occurs between two successive iterations

Hard Clustering Algorithms – k-means (cont)

- Remarks:
  - k-means recovers compact clusters.
  - Sequential versions of the k-means, where the updating of the representatives takes place immediately after the identification of the representative that lies closer to the current input vector $\underline{x}_i$, have also been proposed.
  - A variant of the k-means results if the number of vectors in each cluster is constrained *a priori*.
  - The computational complexity of the k-means is $O(Nmq)$, where q is the number of iterations required for convergence. In practice, m and q are significantly less than N, thus, k-means becomes eligible for processing large data sets.

- Further remarks:

  Some drawbacks of the original k-means accompanied with the variants of the k-means that deal with them are discussed next.

❖ Hard Clustering Algorithms – k-means (cont)

➢ Drawback 1: *Different initial partitions may lead k-means to produces different final clusterings, each one corresponding to a different local minimum.*

Strategies for facing drawback 1:

- Single run methods
  - Use a sequential algorithm (discussed previously) to produce initial estimates for $\underline{\theta}_j$'s.
  - Partition randomly the data set into m subsets and use their means as initial estimates for $\underline{\theta}_j'$ s.
- Multiple run methods
  - Create different partitions of X, run k-means for each one of them and select the best result.
  - Compute the representatives iteratively, one at a time, by running k-means mN times. It is claimed that convergence is independent of the initial estimates of $\underline{\theta}_j'$ s.
- Utilization of tools from stochastic optimization techniques (simulated annealing, genetic algorithms etc).

❖ Hard Clustering Algorithms – k - means (cont)

➢ Drawback 2: *Knowledge of the number of clusters m is required a priori.*

Strategies for facing drawback 2:

- Employ splitting, merging and discarding operations of the clusters resulting from k-means.
- Estimate m as follows:
  - Run a sequential algorithm many times for different thresholds of dissimilarity Θ.
  - Plot Θ versus the number of clusters and identify the largest plateau in the graph and set m equal to the value that corresponds to this plateau.

❖ Hard Clustering Algorithms – k - means (cont)

- Drawback 3: *k-means is sensitive to outliers and noise.*

  Strategies for facing drawback 3:

  - Discard all "small" clusters (they are likely to be formed by outliers).
  - Use a k-medoids algorithm (see below), where a cluster is represented by one of its points.

- Drawback 4: *k-means is not suitable for data with nominal (categorical) coordinates.*

  Strategies for facing drawback 4:

  - Use a k-medoids algorithm.

❖ Hard Clustering Algorithms

➢ *k-Medoids Algorithms*

- Each cluster is represented by a vector selected among the elements of X (medoid).
- A cluster contains
  - Its medoid
  - All vectors in X that
    - Are not used as medoids in other clusters
    - Lie closer to its medoid than the medoids representing other clusters.

Let Θ be the set of medoids of all clusters, I_Θ the set of indices of the points in X that constitute Θ and $I_{X-\Theta}$ the set of indices of the points that are not medoids.

- Obtaining the set of medoids Θ that best represents the data set, X is equivalent to minimizing the following cost function

➢ k-Medoids Algorithms (cont)

$$J(\Theta, U) = \sum_{i \in I_{X-\Theta}} \sum_{j \in I_{\Theta}} u_{ij} d(\underline{x}_i, \underline{x}_j)$$

with

$$u_{ij} = \begin{cases} 1, & if \;\; d(\underline{x}_i, \underline{x}_j) = \min_{q \in I_{\Theta}} d(\underline{x}_i, \underline{x}_q) \\ 0, & otherwise \end{cases}, \quad i = 1, \ldots, N$$

- Representing clusters with mean malues vs representing clusters with medoids

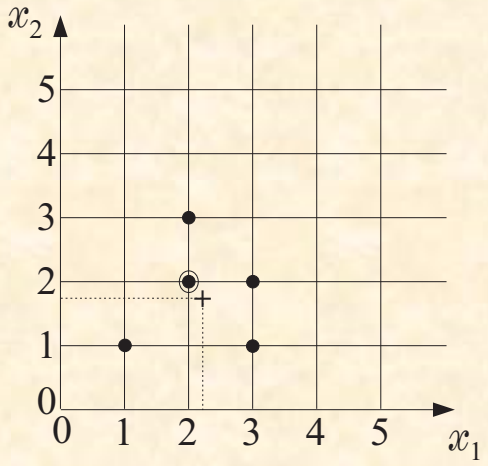
Mean Values	Medoids
1. Suited only for continuous domains	1. Suited for either cont. or discrete domains
2. Algorithms using means are sensitive to outliers	2. Algorithms using medoids are less sensitive to outliers
3. The mean possess a clear geometrical and statistical meaning	3. The medoid has not a clear geometrical meaning
4. Algorithms using means are not computationally demanding	4. Algorithms using medoids are more computationally demanding

➢ k-Medoids Algorithms (cont)

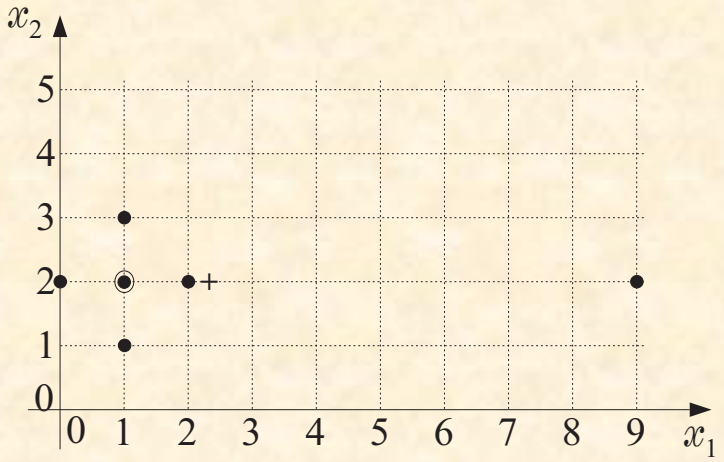
➢ Example 7: (It illustrates the first two points in the above comparison)

(a) The five-point two-dimensional set stems from the discrete domain $D=\{1,2,3,4,\ldots\}x\{1,2,3,4,\ldots\}$. Its medoid is the circled point and its mean is the "+" point, which does not belong to D.

(b) In the six-point two-dimensional set , the point $(9,2)$ can be considered as an outlier. While the outlier affects significantly the mean of the set, it does not affect its medoid.



(a)



(b)