Clustering: k-Means and k-Medoids

Razvan C. Bunescu
Department of Computer Science @ CCI
rbunescu@uncc.edu
Unsupervised Learning: Clustering

- Partition unlabeled examples into disjoint clusters such that:
  - Examples in the same cluster are very similar.
  - Examples in different clusters are very different.
Unsupervised Learning: Clustering

- Partition unlabeled examples into disjoint clusters such that:
  - Examples in the same cluster are very similar.
  - Examples in different clusters are very different.
Divisive Clustering with $k$-Means

- The goal is to produce $k$ clusters $C = \{C_1, C_2, \ldots, C_k\}$ such that instances are close to the cluster centroids:
  - The cluster centroid $m_i$ is the mean of all instances in the cluster $C_i$.

- Optimization problem:

$$\hat{C} = \arg \min_C J(C)$$

$$J(C) = \sum_{i=1}^{k} \sum_{x \in C_i} \| x - m_i \|^2$$
The $k$-Means Algorithm

1. start with some seed centroids $\mathbf{m}_1^{(0)}, \mathbf{m}_2^{(0)}, \ldots, \mathbf{m}_k^{(0)}$
2. set $t \leftarrow 0$.
3. while not converged:
4. for each $\mathbf{x}$:
5. set $\mathbf{m}^{(t)}(\mathbf{x}) \leftarrow \arg\min_{\mathbf{m}_i^{(t)}} \| \mathbf{x} - \mathbf{m}_i^{(t)} \|$ \hspace{1cm} [E] step
6. set $C_i^{(t+1)} \leftarrow \{ \mathbf{x} \mid \mathbf{m}^{(t)}(\mathbf{x}) = \mathbf{m}_i^{(t)} \}$
7. set $\mathbf{m}_i^{(t+1)} \leftarrow \frac{1}{|C_i^{(t+1)}|} \sum_{\mathbf{x} \in C_i^{(t+1)}} \mathbf{x}$ \hspace{1cm} [M] step
8. set $t \leftarrow t + 1$
The $k$-Means Algorithm ($k = 2$)

- Pick seeds
- Reassign clusters
- Compute centroids
- Reassign clusters
- Compute centroids
- Reassign clusters
- Converged!
The $k$-Means Algorithm ($k = 2$)
The \textit{k-Means Algorithm} ($k = 2$)
The $k$-Means Algorithm ($k = 2$)
The $k$-Means Algorithm ($k = 2$)
The \textit{k}-Means Algorithm \((k = 2)\)
The $k$-Means Algorithm ($k = 2$)
The $k$-Means Algorithm ($k = 2$)
The $k$-Means Algorithm ($k = 2$)
The $k$-Means Algorithm ($k = 2$)
The $k$-Means Algorithm

- The objective function monotonically decreases at every iteration:
  \[ J^{(t)} \geq J^{(t+1)} \]
The \( k \)-Means Algorithm

- Optimization problem is NP-hard:
  - Results depend on seed selection.
  - Improve performance by providing \textit{must-link} and/or \textit{cannot-link} constraints \( \Rightarrow \) \textit{semi-supervised clustering}.

- Time complexity for each iteration is \( O(knm) \):
  - number of clusters is \( k \).
  - feature vectors have dimensionality \( m \).
  - total number of instances is \( n \).
The $k$-Means Algorithm

1. start with some seed centroids $m_1^{(0)}, m_2^{(0)}, \ldots, m_k^{(0)}$
2. set $t \leftarrow 0.$
3. while not converged:
4. for each $x$:
5. set $m^{(t)}(x) \leftarrow \arg \min_{m_i^{(t)}} \| x - m_i^{(t)} \|$ \hspace{1cm} \text{[E] step}
6. set $C_i^{(t+1)} \leftarrow \{ x \mid m^{(t)}(x) = m_i^{(t)} \}$
7. set $m_i^{(t+1)} \leftarrow \frac{1}{|C_i^{(t+1)}|} \sum_{x \in C_i^{(t+1)}} x$ \hspace{1cm} \text{[M] step}
8. set $t \leftarrow t + 1$
The \( k \)-Medoids Algorithm

1. start with some random seed centroids \( m_1^{(0)}, m_2^{(0)}, \ldots, m_k^{(0)} \)
2. set \( t \leftarrow 0 \).
3. while not converged:
   4. for each \( x \):
      5. set \( m^{(t)}(x) \leftarrow \arg \min_{m_i^{(t)}} d(x - m_i^{(t)}) \) \([E]\) step
   6. set \( C_i^{(t+1)} \leftarrow \{ x \mid m^{(t)}(x) = m_i^{(t)} \} \)
   7. set \( m_i^{(t+1)} \leftarrow \arg \min_{x \in C_i^{(t+1)}} \sum_{y \in C_i^{(t+1)}} d(x, y) \) \([M]\) step
8. set \( t \leftarrow t + 1 \)