CS522 Advanced Database Systems
Clustering: K-Means

Chengyu Sun
California State University, Los Angeles

K-Means
- Input: dataset \( D \) and number of clusters \( k \)
- Algorithm
  1. Randomly choose \( k \) objects as cluster centers
  2. Assign each object to the closest cluster center
  3. Update each cluster center
  4. Repeat 2 until there is no reassignment occurs

K-Means Example

Key Issues in K-Means
- Distance measure?
  - Euclidean, Manhattan, Cosine ...
- Cluster center?
  - Mean, median

Need for Objective Function

The best clustering is the one that minimize the "errors" defined by an objective function

Notations

<table>
<thead>
<tr>
<th>( D )</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k )</td>
<td>The number of clusters</td>
</tr>
<tr>
<td>( C_i )</td>
<td>The center of the ( i )th cluster</td>
</tr>
<tr>
<td>( x )</td>
<td>An object</td>
</tr>
</tbody>
</table>
**Objective Functions**

Sum of the Squared Error (SSE):

\[ \text{SSE} = \sum_{i=1}^{k} \sum_{c \in C} \text{dist}_{i}(x, c)^2 \]

Sum of the Absolute Error (SAE):

\[ \text{SAE} = \sum_{i=1}^{k} \sum_{c \in C} |\text{dist}_{i}(x, c)| \]

**Minimize an Object Function**

- **Example:**
  - One dimensional data
  - One cluster
  - SSE

\[ \text{SSE}(c) = \sum_{x \in C} (c - x)^2 \implies \frac{\partial}{\partial c} \text{SSE}(c) = 0 \]

**Distances, Centroids, and Objective Functions**

<table>
<thead>
<tr>
<th>Distance Function</th>
<th>Centroid</th>
<th>Objective Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manhattan (L₁)</td>
<td>Median</td>
<td>Sum of L₁ distance</td>
</tr>
<tr>
<td>Squared Euclidean (L₂)</td>
<td>Mean</td>
<td>Sum of squared L₂ distance</td>
</tr>
<tr>
<td>Cosine</td>
<td>Mean</td>
<td>Sum of cosine distance</td>
</tr>
<tr>
<td>Bregman Divergence</td>
<td>Mean</td>
<td>Sum of Bregman divergence</td>
</tr>
</tbody>
</table>

**Dealing with the Problem of Initial Centroid Selection**

- Perform several runs of K-Means and select the clustering with the smallest SSE
- What’s the probability of picking K objects, each from a different cluster??
- Use a hierarchical clustering algorithm on a sample to get K initial clusters
- Select centroid one by one, and each one is the farthest away from previously selected ones

**Another K-Means Example ...**

- Iteration 5
- Iteration 6
- Iteration 7
- Iteration 8

**... Another K-Means Example**

- Iteration 1
- Iteration 2
- Iteration 3
- Iteration 4
Postprocessing

- Escape local SSE minimum by performing alternate clustering splitting and merging

Postprocessing – Splitting

- Splitting the cluster with the largest SSE on the attribute with the largest variance
- Introduce another centroid
  - The point that is farthest from current centroids
  - Randomly chosen

Postprocessing – Merging

- Disperse a cluster and reassign its objects
- Merge two clusters that are closest to each other

Bisecting K-Means

1. Initialize a list of clusters with one cluster containing all the objects
2. Choose one cluster from the list
3. Split the cluster into two using basic K-Means, and add them back to the list
4. Repeat Step 2 until $k$ clusters are reached
5. Perform one more basic K-Means using the centroids of the $k$ clusters as initial centroids

About Bisecting K-Means

- Step 2
  - Choose the largest cluster
  - Choose the cluster with the largest SSE
- Step 3
  - Perform basic K-Means several times and choose the clustering with the smallest SSE
  - Less susceptible to initialization problems
  - Why??

Handling Empty Clusters

- Choose a replacement centroid
  - The point that’s farthest away from any current centroid
  - A point from the cluster with the highest SSE
### K-Medoids

- Instead of using mean/centroid, use medoid, i.e., representative object
- Objective function: sum of the distances of the objects to their medoid
- Differ from K-Means in how the medoids are updated

### PAM (Partition Around Medoids)

1. Randomly choose $k$ objects as initial medoids
2. Assign each object to its closest medoid
3. For each non-medoid object $x$:
   - For each medoid $c_i$, calculate the reduction of the total distance if $c_i$ is replaced by $x$
4. Replace the $c_i$ with $x$ that results in maximum total distance reduction
5. Repeat Step 2 until the total distance cannot be reduced

### PAM Example

![PAM Example Diagram]

### K-Means vs. K-Medoids

- Requires the notion of mean/centroid
- More susceptible to outliers
- $O(k(n-k))$ per iteration
- Works for all distance measures
- Less susceptible to outliers
- $O(k(n-k)^2)$ per iteration

### Limitations of K-Means – Different Types of Clusters

- Continuity-based
- Density-based

### Limitations of K-Means – Differing Sizes

- Continuity-based
- Density-based
Limitations of K-Means – Non-globular Shapes

Readings

Textbook 10.2