Clustering Algorithms: A Quick Overview of k-means, k-median, and k-medoids

V.S. Subrahmanian

Fall 2013

What is Clustering?

- You have a set **S** of vectors in an *n*-dimensional space.
- You have an integer $k > 0$.
- You want to split **S** into *k* buckets such that the "intra-bucket" distance is small, i.e. items within a bucket are close to each other according to some distance measure.

Clustering Approaches

- Clustering approaches vary depending upon what distance metric is used, e.g.
  – Euclidean distance vs.
  – Manhattan distance.
- Well known clustering methods include:
  – *K*-means
  – *K*-median
  – *K*-medoids (Partitioning around medoids – PAM)

K-Means

- Suppose we split **S** into *k* buckets $S_1,..,S_k$ such that:
  - Each bucket S_i has a center c_i;
  - $Cost(S_1,..,S_k) = \sum_{i=1}^{k} \sum_{x \ in \ S_i} (x - c_i)^2$
- We want to split **S** into *k* buckets $S_1,..,S_k$ such that the cost is minimized.
- Problem is NP-hard.
- In practice, a heuristic algorithm is used.

K-Means Heuristic Algorithm

1. Split **S** into an initial set of k buckets $S_1,..,S_k$.
2. Compute the mean m_i of each cluster.
3. For each element s ε **S**
   – Find the bucket S_i such that $|m_i - s|^2$ is minimal.
   – Put s in that bucket.
4. Repeat steps (2) and (3) till no change occurs.

K-Median Heuristic Algorithm

1. Split **S** into an initial set of k buckets $S_1,..,S_k$.
2. Compute the median m_i of each cluster.
3. For each element s ε **S**
   - Find the bucket S_i such that $|m_i - s|$ is minimal.
   - Put s in that bucket.
4. Repeat steps (2) and (3) till no change occurs.

K-Means Example in one dimension

- **S** = { 1,2,3,4,5,11,12,13,14,16}
- *k = 2.*
- Initially, we choose a random partition.

S_1 = {1,2,3,5,12}
S_2 = {4,11,13,14,16}

K-Means Example in on dimension

- Find the means of each bucket.

S_1 = {1,2,3,5,12}; m1 = 23/5 = 4.6
S_2 = {4,11,13,14,16}; m2 = 58/5 = 11.6

K-Means Example in on dimension

- Consider each element and move it to the bucket whose mean it is closer to.

S_1 = {1,2,3,5,4};
S_2 = {12,11,13,14,16};

K-Means Example in on dimension

- Consider each element and move it to the bucket whose mean it is closer to.

$S_1 = \{1,2,3,5,4\};$
$S_2 = \{12,11,13,14,16\};$

First iteration of the loop is finished.

K-Means Example in on dimension

- Recompute means

S_1 = {1,2,3,5,4}; m1 = 3
S_2 = {12,11,13,14,16}; m2 = 13

K-Means Example in one dimension

- Move elements to buckets whose centers it is "closest" to.

S_1 = {1,2,3,5,4}; m1 = 3
S_2 = {12,11,13,14,16}; m2 = 13

No change occurs. Done !

In-class exercise

- K-means in 2-dimensions
- **S** = { (1,2), (1,3), (2,3),(1,1), (7,8),(7,7), (6,8),(8,7)}

k-medians in one dimension

- **S** = { 1,2,3,4,5,11,12,13,14,16}
- *k = 2.*

S_1 = {1,2,3,5,12}
S_2 = {4,11,13,14,16}

Pick an initial partition

k-medians in one dimension

- **S** = { 1,2,3,4,5,11,12,13,14,16}
- *k = 2.*

S_1 = {1,2,3,5,12}; m1= 3
S_2 = {4,11,13,14,16}; m2= 13

Find medians of each partition

k-medians in one dimension

- **S** = { 1,2,3,4,5,11,12,13,14,16}
- *k = 2.*

S_1 = {1,2,3,5,12}; m1= 3
S_2 = {4,11,13,14,16}; m2= 13

Move elements *s* to bucket whose median is closest to *s*

In-class Exercise: k-medians in one dimension

- **S** = { 1,2,3,4,5,11,12,13,14,16}
- *k = 2.*

S_1 = {1,2,3,5,4}; m1= 3
S_2 = {12,11,13,14,16}; m2= 13

End of first iteration

In-class Exercise: k-medians in one dimension

- **S** = { 1,2,3,4,5,11,12,13,14,16}
- *k = 2.*

S_1 = {1,2,3,5,4}; m1= 3
S_2 = {12,11,13,14,16}; m2= 13

No change at the end of the 2nd iteration. Done!

In-class exercise

- K-medians in 2-dimensions
- **S** = { (1,2), (1,3), (2,3),(1,1), (7,8),(7,7), (6,8),(8,7)}

K-Medoid Heuristic Algorithm

1. Split **S** into an initial set of k buckets $S_1,..,S_k$.
2. Compute the "medoid" m_i of each bucket S_i. This is the element m_i <u>contained inside</u> bucket S_i s.t. $\sum_{s \ in \ S_i} d(m_i, s)$ is minimized.
3. For each element s ε **S**
   – Find the bucket S_i such that $d(m_i, s)$ is minimal.
   – Put s in that bucket.
4. Repeat steps (2) and (3) till no change occurs.

Medoid May Differ from Median

- Let S = {1,2,3,20,100}.
- Median is 3, but medoid is 20. Why?

m	$\sum_{s\ in\ S_i} d(m_i, s)$
1	*10167*
2	*9930*
3	*9703*
20	*7374*

In-class Exercise: k-medoids in one dimension

- **S** = { 1,2,3,4,5,11,12,13,14,16}
- *k = 2.*

In-class exercise

- K-medoids in 2-dimensions
- **S** = { (1,2), (1,3), (2,3),(1,1), (7,8),(7,7), (6,8),(8,7)}