k-median Clustering

TOP: Data Clustering 076/091

Instructor: Sayan Bandyapadhyay

Portland State University

Outline

A Robust Estimator

- k-center is very sensitive to outliers
- A quantity that is robust/not very sensitive: median
- Need to corrupt several data points to significantly change the median

A Robust Estimator

- k-center is very sensitive to outliers
- A quantity that is robust/not very sensitive: median
- Need to corrupt several data points to significantly change the median
- Hard to define in higher dimension
- Generalize another definition of the median

Another definition of median

Given a set X of numbers, median m minimizes the total/average distance

$$\sum_{p \in X} ||p - c||$$

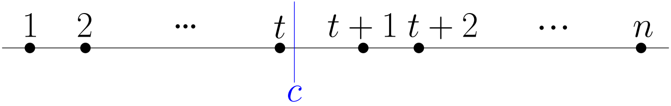
over all $c \in \mathbb{R}$.

Another definition of median

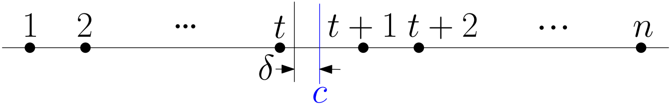
Given a set X of numbers, median m minimizes the total/average distance

$$\sum_{p \in X} ||p - c||$$

over all $c \in \mathbb{R}$.

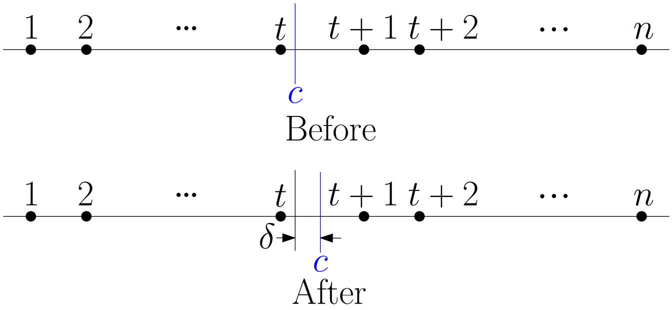


Before



After

Change in cost



- Change is $t \cdot \delta - (n - t)\delta = (2t - n)\delta$
- It is -ve as long as $t < n/2$
- It is 0 if $n/2$ points on both sides of c

k-median Clustering

Given a set X of n points in the metric space $(\mathcal{U}, d)$

- Find a set C of k points (cluster centers) in $\mathcal{U}$ that minimizes,

$$\text{cost}(C) = \sum_{p} d(p, \mathit{NearestCenter}(p))$$

The centers in C are called medoids.

Discrete k-median Clustering

Given a set X of n points in the metric space (X, d)

- Find a set C of k points (cluster centers) in X that minimizes,

$$\mathrm{cost}(C) = \sum_{p} d(p, NearestCenter(p))$$

Outline

An Algorithm for k-median (Arya et al. 2001)

Algorithm Local Search

Require: Set of points X

Start with medoids $M = \{m_1, \ldots, m_k\}$ chosen arbitrarily from X
repeat
    change $\leftarrow$ false
    for $x \in X \setminus M, m \in M$ **do**
        $M' \leftarrow (M \setminus \{m\}) \cup \{x\}$
        if cost$(M') <$ cost(M) **then**
            $M \leftarrow M'$, change $\leftarrow$ true, Break.
        end if
    end for
until change is false
return M

Time Complexity of Local Search

Algorithm Local Search

Require: Set of points X

Start with medoids $M = \{m_1, \ldots, m_k\}$ chosen arbitrarily from X
repeat
  change $\leftarrow$ false
  for $x \in X \setminus M, m \in M$ **do**
    $M' \leftarrow (M \setminus \{m\}) \cup \{x\}$
    if cost(M') $<$ cost(M) **then**
      $M \leftarrow M'$, change $\leftarrow$ true, Break.
    end if
  end for
until change is false
return M

- Neighborhood size is $O(nk)$: cost computation $O(nk)$; Total $O(n^2k^2)$

Time Complexity of Local Search

Algorithm Local Search

Require: Set of points X

Start with medoids $M = \{m_1, \ldots, m_k\}$ chosen arbitrarily from X
repeat
  change $\leftarrow$ false
  for $x \in X \setminus M, m \in M$ **do**
    $M' \leftarrow (M \setminus \{m\}) \cup \{x\}$
    if cost(M') $<$ cost(M) **then**
      $M \leftarrow M'$, change $\leftarrow$ true, Break.
    end if
  end for
until change is false
return M

- Neighborhood size is $O(nk)$: cost computation $O(nk)$; Total $O(n^2k^2)$
- Complexity depends on the repeat-until loop: rate of cost decrement

Changing the cost checking condition

Algorithm Local Search

Require: Set of points X

Start with medoids $M = \{m_1, \ldots, m_k\}$ chosen arbitrarily from X
repeat
  change $\leftarrow$ false
  for $x \in X \setminus M, m \in M$ **do**
    $M' \leftarrow (M \setminus \{m\}) \cup \{x\}$
    if cost$(M') <$ cost$(M)/2$ **then**
      $M \leftarrow M'$, change $\leftarrow$ true, Break
    end if
  end for
until change is false
return M

- Overall ℓ iterations: sets of medoids $M_1, M_2, \ldots, M_\ell$
- Exponential Decrease: cost$(M_\ell) <$ cost$(M_{\ell-1})/2 <$ cost$(M_{\ell-2})/2^2 < \ldots <$ cost$(M_1)/2^{\ell-1}$

Time Complexity

- Overall ℓ iterations: sets of medoids $M_1, M_2, \ldots, M_\ell$
- Exponential Decrease: $\text{cost}(M_\ell) < \text{cost}(M_1)/2^{\ell-1}$

Time Complexity

- Overall ℓ iterations: sets of medoids $M_1, M_2, \ldots, M_\ell$
- Exponential Decrease: $\text{cost}(M_\ell) < \text{cost}(M_1)/2^{\ell-1}$
- $\text{cost}(M_\ell) \geq \text{OPT}$

Time Complexity

- Overall ℓ iterations: sets of medoids $M_1, M_2, \ldots, M_\ell$
- Exponential Decrease: $\mathrm{cost}(M_\ell) < \mathrm{cost}(M_1)/2^{\ell-1}$
- $\mathrm{cost}(M_\ell) \geq \mathrm{OPT}$
- If $\mathrm{Cost}(M_1) \leq n \cdot \mathrm{OPT}$, $\ell = O(\log_2 n)$

Time Complexity

- Overall ℓ iterations: sets of medoids $M_1, M_2, \ldots, M_\ell$
- Exponential Decrease: $\text{cost}(M_\ell) < \text{cost}(M_1)/2^{\ell-1}$
- $\text{cost}(M_\ell) \geq \text{OPT}$
- If $\text{Cost}(M_1) \leq n \cdot \text{OPT}$, $\ell = O(\log_2 n)$
- Complexity is $O(n^2 k^2 \log n)$

A $2n$-approximation for k-median

Compute a 2-approximation for k-center, and use these centers C as medoids

A $2n$-approximation for k-median

Compute a 2-approximation for k-center, and use these centers C as medoids

- OPT-k-center cost $\leq$ OPT-k-median cost (longest distance vs total sum) – Ineq (1)

A $2n$-approximation for k-median

Compute a 2-approximation for k-center, and use these centers C as medoids

- OPT-k-center cost $\leq$ OPT-k-median cost (longest distance vs total sum) – Ineq (1)
- Now, analyze the k-median cost of C
- k-median-cost$(C) \leq n \cdot 2\cdot$ OPT-k-center cost (pay $2\cdot$OPT-k-center n times)

A $2n$-approximation for k-median

Compute a 2-approximation for k-center, and use these centers C as medoids

- OPT-k-center cost $\leq$ OPT-k-median cost (longest distance vs total sum) – Ineq (1)
- Now, analyze the k-median cost of C
- k-median-cost$(C) \leq n \cdot 2\cdot$ OPT-k-center cost (pay $2\cdot$OPT-k-center n times)
- Then, by Ineq (1),
  k-median-cost$(C) \leq n \cdot 2\cdot$OPT-k-median cost

Analysis of Local Search

- What is the approximation factor?

Analysis of Local Search

- What is the approximation factor? 5
- If we swap 2 medoids, 4-approximation
- If we swap p medoids, $(3 + 2/p)$-approximation
- But, time complexity $O(n^{p+1} k^{p+1} \log n)$
- The analyses are beyond the scope