COL870: Clustering Algorithms

Ragesh Jaiswal, CSE, IIT Delhi

Streaming and Online Clustering

- Online Setting: In this computational setting the data is accessed as an *endless stream*. At every time step t, an online algorithm should be prepared to output a solution that is good with respect to the input points seen until time t.

Online algorithm template for k-means/median/center

- Repeat forever:
  - Get a new data point x
  - Update the current set of k centers

- Is it possible to output optimal centers at every time step?

Online Clustering

The current material is from Sanjoy Dasgupta's lecture notes.

- Online Setting: In this computational setting the data is accessed as an *endless stream*. At every time step t, an online algorithm should be prepared to output a solution that is good with respect to the input points seen until time t.

Online algorithm template for k-means/median/center

- Repeat forever:
  - Get a new data point x
  - Update the current set of k centers

- Is it possible to output optimal centers at every time step?
- We will see two online algorithms for the k-center problem that gives an approximation factor of 8 at every time step (for any metric space).

Online Algorithm for k-center
Algorithm #1

Algorithm #1 (Doubling Algorithm) [CCFM97]

- $T \leftarrow \{\text{first } k \text{ distinct data points}\}$
- $R \leftarrow$ smallest interpoint distance in T
- Repeat forever:
    - while $|T| \leq k$:
        - **(A)** Get a new point x
        - If $D(x, T) > 2R$ then $T \leftarrow T \cup \{x\}$
    - **(B)** $T' \leftarrow \{\}$
    - while there exists $z \in T$ such that $D(z, T') > 2R$
        - $T' \leftarrow T' \cup \{z\}$
    - $T \leftarrow T'$
    - **(C)** $R \leftarrow 2R$

- Claim 1: All data points seen so far are (i) within distance $2R$ of T at **(B)** and (ii) within distance $4R$ of T at **(C)**.

Online Clustering

The current material is from Sanjoy Dasgupta's lecture notes.

Algorithm #1 (Doubling Algorithm) [CCFM97]

- $T \leftarrow \{\text{first } k \text{ distinct data points}\}$
- $R \leftarrow$ smallest interpoint distance in T
- Repeat forever:
    - while $|T| \leq k$:
        - **(A)** Get a new point x
        - If $D(x, T) > 2R$ then $T \leftarrow T \cup \{x\}$
    - **(B)** $T' \leftarrow \{\}$
    - while there exists $z \in T$ such that $D(z, T') > 2R$
        - $T' \leftarrow T' \cup \{z\}$
    - $T \leftarrow T'$
    - **(C)** $R \leftarrow 2R$

- Claim 1: All data points seen so far are (i) within distance $2R$ of T at **(B)** and (ii) within distance $4R$ of T at **(C)**.
- Claim 2: At **(B)**, there are $k + 1$ centers at distance $\geq R$ from each other.

Algorithm #1 (Doubling Algorithm) [CCFM97]

- $T \leftarrow$ {first k distinct data points}
- $R \leftarrow$ smallest interpoint distance in T
- Repeat forever:
    - while $|T| \leq k$:
        - **(A)** Get a new point x
        - If $D(x, T) > 2R$ then $T \leftarrow T \cup \{x\}$
    - **(B)** $T' \leftarrow \{\}$
    - while there exists $z \in T$ such that $D(z, T') > 2R$
        - $T' \leftarrow T' \cup \{z\}$
    - $T \leftarrow T'$
    - **(C)** $R \leftarrow 2R$

- Claim 1: All data points seen so far are (i) within distance $2R$ of T at **(B)** and (ii) within distance $4R$ of T at **(C)**.
- Claim 2: At **(B)**, there are $k + 1$ centers at distance $\geq R$ from each other.
- Claim 3: Whenever the algorithm is at **(A)**, $cost(T) \leq 8 \cdot cost(\text{cost of optimal } k \text{ centers for data seen so far})$.

Online Algorithm for k-center
Algorithm #2

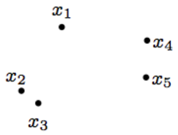
- Suppose, we would like the online algorithm to be prepared to output a good solution at every time step for *all* values of k.
- We define a new data structure called a *cover tree* for the given data points $x_1, ..., x_n$ that will be used in the algorithm. We will currently assume that $D(x_i, x)_j) \leq 1$ for all i, j.

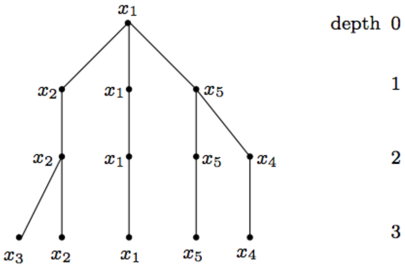
Cover Tree

- Each node of the tree is associated with one of the data points x_i.
- If a node is associated with x_i, then one of its children must also be associated with x_i.
- All nodes at depth j are at distance at least $\frac{1}{2^j}$ from each other.
- Each node at depth $j + 1$ is within distance $\frac{1}{2^j}$ of its parent.

Cover Tree

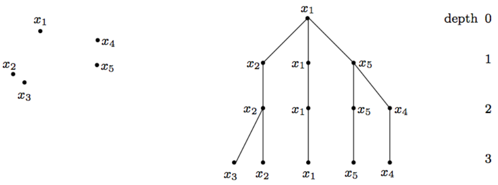
- Each node of the tree is associated with one of the data points x_i.
- If a node is associated with x_i, then one of its children must also be associated with x_i.
- All nodes at depth j are at distance at least $\frac{1}{2^j}$ from each other.
- Each node at depth $j+1$ is within distance $\frac{1}{2^j}$ of its parent.





Cover Tree

- Each node of the tree is associated with one of the data points x_i.
- If a node is associated with x_i, then one of its children must also be associated with x_i.
- All nodes at depth j are at distance at least $\frac{1}{2^j}$ from each other.
- Each node at depth $j+1$ is within distance $\frac{1}{2^j}$ of its parent.



- For any k, consider the deepest level of the tree with $\leq k$ nodes, and let T_k be those nodes. Then $cost(T_k) \leq 8 \cdot cost(\text{optimal } k \text{ centers})$

An Online Algorithm for k-means

- Here is an online algorithm for the k-means problem that is used in practice.

Online k-means

- Initialise the k cenerts $t_1, ..., t_k$ in any manner
- Create counters $n_1, ..., n_k$, all initialised to 0
- Repeat forever:
  - get data point x
  - Let t_i be its closest centre
  - Set $t_i \leftarrow \frac{n_i t_i + x}{n_i + 1}$ and $n_i \leftarrow n_i + 1$

Streaming Algorithms

- Streaming algorithms is expected to process a finite amount of data as opposed to online algorithms that are supposed to run forever.
- Here are the other key differences:

Online setting	**Streaming setting**
- Endless stream of data	- Stream of (known) length n
- Fixed amount of memory	- Memory available is $o(n)$
- Tested at every time step	- Tested only at the very end
- Each point is seen only once	- More than one pass may be possible

- Note that we have already seen a streaming algorithm for the k-median problem in any metric space that use n^{ε} memory and give an approximation factor of $c^{1/\varepsilon}$.

End