Coresets for k-center Clustering

TOP: Data Clustering 076/091

Instructor: Sayan Bandyapadhyay

Portland State University

Outline

Coresets

Summary of the dataset

Coresets

Summary of the dataset

- Subset of the original set

Coresets

Summary of the dataset

- Subset of the original set
- Has size much smaller than n (often independent of n)

Coresets

Summary of the dataset

- Subset of the original set
- Has size much smaller than n (often independent of n)
- A subset of points good enough for clustering

Coresets

Summary of the dataset

- Subset of the original set
- Has size much smaller than n (often independent of n)
- A subset of points good enough for clustering

Advantages

- Makes clustering much faster

Coresets

Summary of the dataset

- Subset of the original set
- Has size much smaller than n (often independent of n)
- A subset of points good enough for clustering

Advantages

- Makes clustering much faster
- Small space complexity: suitable for distributed and streaming setting

Outline

Euclidean k-center

Given a set X of n points in $\mathbb{R}^d$

- Find a set C of k balls (clusters) in $\mathbb{R}^d$ that contains all the points in X and minimizes,

$$\mathrm{cost}(C) = \max_{c \in C} \mathrm{radius}(c)$$

Euclidean k-center

Given a set X of n points in $\mathbb{R}^d$

- Find a set C of k balls (clusters) in $\mathbb{R}^d$ that contains all the points in X and minimizes,

$$\mathrm{cost}(C) = \max_{c \in C} \mathrm{radius}(c)$$

(ϵ-expansion.) An ϵ-expansion of a clustering C is a new set of balls obtained from C by expanding the radii of the balls in C by $\epsilon \cdot cost(C)$

Euclidean k-center

Given a set X of n points in $\mathbb{R}^d$

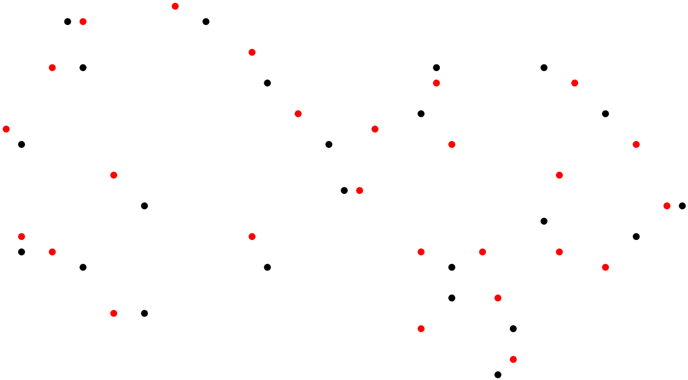
- Find a set C of k balls (clusters) in $\mathbb{R}^d$ that contains all the points in X and minimizes,

$$\text{cost}(C) = \max_{c \in C} \text{radius}(c)$$

(ϵ-expansion.) An ϵ-expansion of a clustering C is a new set of balls obtained from C by expanding the radii of the balls in C by $\epsilon \cdot cost(C)$

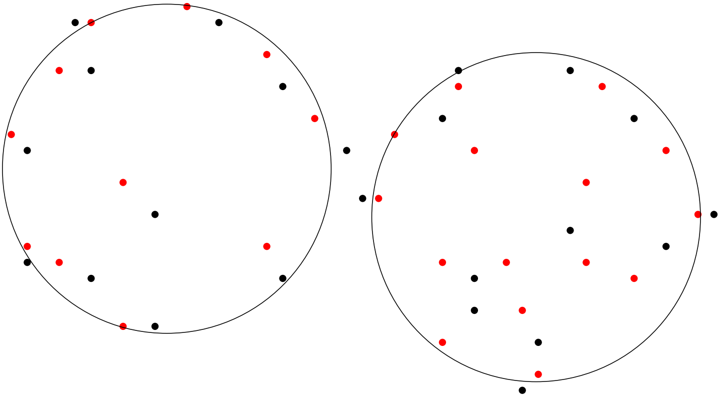
(ϵ-coreset.) A subset $S \subseteq X$ is an ϵ-coreset if for any clustering C of S, the ϵ-expansion of C contains X

An Example Coreset

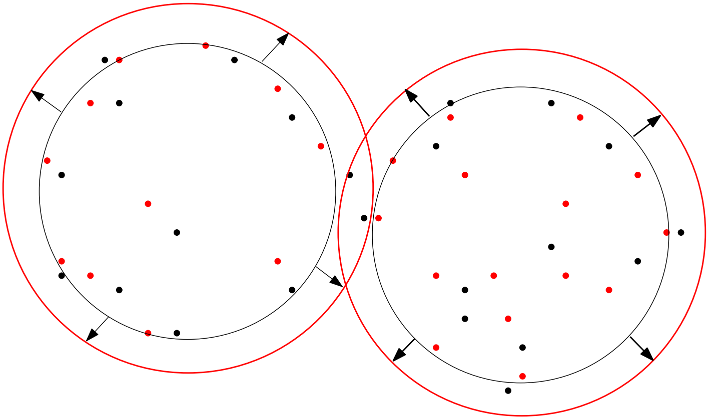


Red points form the coreset

A Clustering of the Coreset



ϵ-expansion of the Clusters



Outline

An Algorithm

- For the time being, assume that we know the optimal clusters/balls O

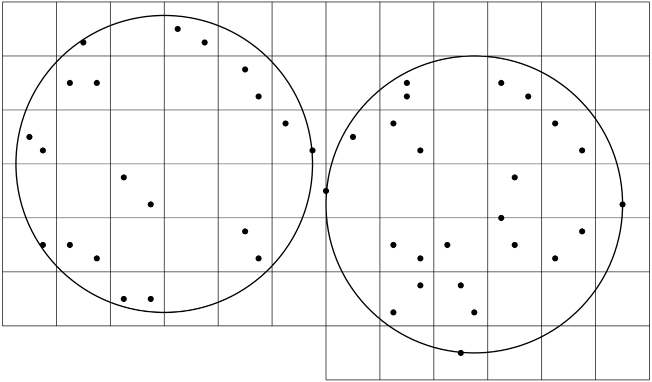
An Algorithm

- For the time being, assume that we know the optimal clusters/balls O
- For each ball, cover it by an $\epsilon OPT/2d$ length d-dimensional grid

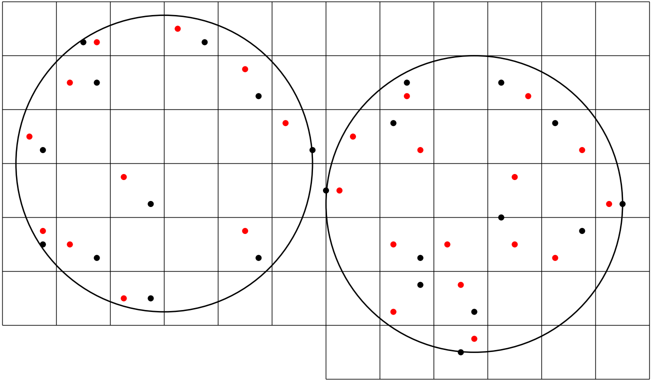
An Algorithm

- For the time being, assume that we know the optimal clusters/balls O
- For each ball, cover it by an $\epsilon OPT/2d$ length d-dimensional grid
- From each non-empty grid cell, pick one point of X, and add it to coreset S

Overlaying the Grid



Selecting Points



Red points are added to coreset

Outline

The Size of the Coreset

- Each OPT radius ball in O is contained in a hypercube of length 2 OPT

The Size of the Coreset

- Each OPT radius ball in O is contained in a hypercube of length 2 OPT
- Overlay an $\epsilon OPT/2d$ length grid on this hypercube

The Size of the Coreset

- Each OPT radius ball in O is contained in a hypercube of length 2 OPT
- Overlay an $\epsilon OPT/2d$ length grid on this hypercube
- The number of cells $(2OPT)^d/(\epsilon OPT/2d)^d = (4d/\epsilon)^d$

The Size of the Coreset

- Each OPT radius ball in O is contained in a hypercube of length 2 OPT
- Overlay an $\epsilon OPT/2d$ length grid on this hypercube
- The number of cells $(2OPT)^d/(\epsilon OPT/2d)^d = (4d/\epsilon)^d$
- Total size $k(4d/\epsilon)^d$ for k balls

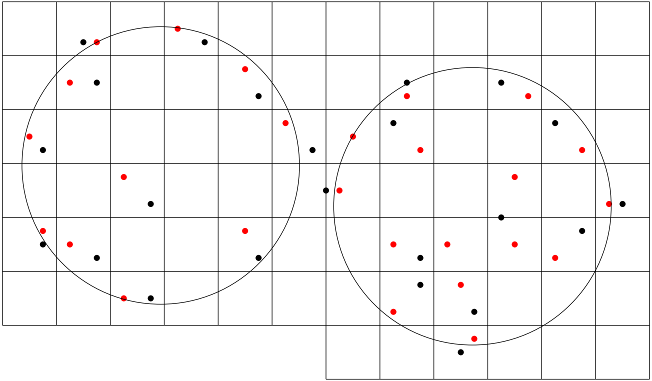
Why S is a Coreset?

- Consider any clustering C of S

Why S is a Coreset?

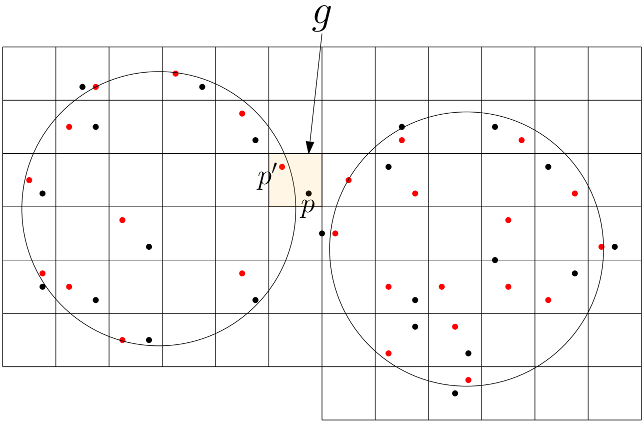
- Consider any clustering C of S
- Need to show: the ϵ-expansion of C contains X

Clustering C of S



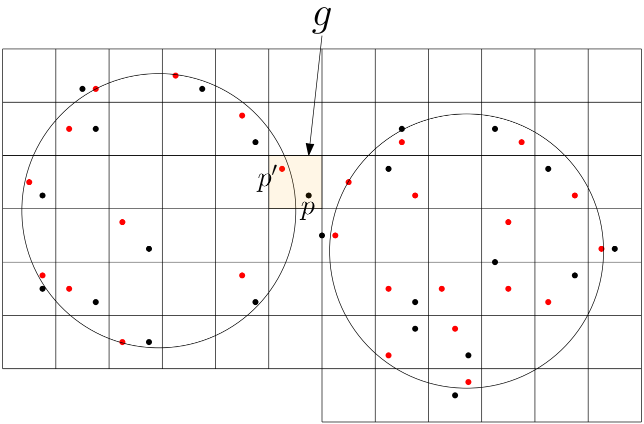
Might not cover all points of X

An Uncovered Point p



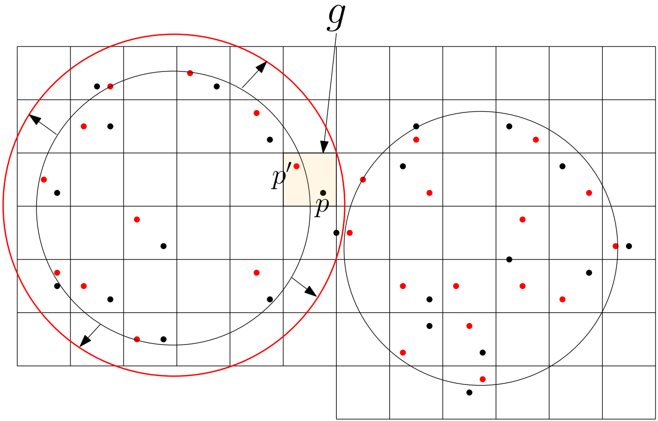
p' must be in the coreset and so covered

How far are p and p'?



$$||p - p'|| \leq \sqrt{d}(\epsilon OPT/2d) \leq (\epsilon/2)OPT$$

ϵ-expansion of C contains X



Expansion by $(\epsilon/2)OPT \leq (\epsilon/2)(1+\epsilon)cost(C) \leq \epsilon cost(C)$ is sufficient to cover X

Why S is a Coreset?

- Consider any clustering C of S
- Need to show: the ϵ-expansion of C contains X
- Consider any point p of X not in the coreset
- Consider the gridcell g that p is in
- p is not in S, so $p' \in g$ is chosen in S
- $||p - p'|| \leq \sqrt{d}(\epsilon OPT/2d) \leq (\epsilon/2)OPT$
- Expansion by $(\epsilon/2)OPT \leq (\epsilon/2)(1 + \epsilon)cost(C) \leq \epsilon cost(C)$ is sufficient to cover X

How to Guess OPT?

How to Guess OPT?

- Compute a 2-approximation A of OPT - Greedy 1/2

How to Guess OPT?

- Compute a 2-approximation A of OPT - Greedy 1/2
- $cost(A) \in [OPT, 2OPT]$ is a good estimator of OPT

How to Guess OPT?

- Compute a 2-approximation A of OPT - Greedy 1/2
- $cost(A) \in [OPT, 2OPT]$ is a good estimator of OPT
- Instead of optimal balls overlay the grid on balls of A

How to Guess OPT?

- Compute a 2-approximation A of OPT - Greedy 1/2
- $cost(A) \in [OPT, 2OPT]$ is a good estimator of OPT
- Instead of optimal balls overlay the grid on balls of A
- Use gridcells of length $\epsilon cost(A)/4d$ instead of $\epsilon OPT/2d$

How to Guess OPT?

- Compute a 2-approximation A of OPT - Greedy 1/2
- $cost(A) \in [OPT, 2OPT]$ is a good estimator of OPT
- Instead of optimal balls overlay the grid on balls of A
- Use gridcells of length $\epsilon cost(A)/4d$ instead of $\epsilon OPT/2d$
- Finer granularity: $\epsilon cost(A)/4d \in [\epsilon OPT/4d, \epsilon OPT/2d]$

Outline

The Setting

- set X of n data points distributed over m machines

The Setting

- set X of n data points distributed over m machines
- Each machine has limited storage of $<< n$

The Setting

- set X of n data points distributed over m machines
- Each machine has limited storage of $<< n$
- Communication between machines is possible

The Setting

- set X of n data points distributed over m machines
- Each machine has limited storage of $<< n$
- Communication between machines is possible
- Machines are synced in rounds

The Setting

- set X of n data points distributed over m machines
- Each machine has limited storage of $<< n$
- Communication between machines is possible
- Machines are synced in rounds
- The goal is to compute a k-center clustering of X - each machine should know the k centers

Approximation via Coresets

Algorithm Distributed k-center

Require: Set of points X_i in machine i for $1 \leq i \leq m$, $\epsilon > 0$

1: Each machine i computes an ϵ-coreset C_i of X_i in round 1
2: Each machine i except 1 sends C_i to machine 1 in round 2
3: Machine 1 computes a k-center clustering of $\cup_i C_i$ and sends the k centers to all machines in round 3

Approximation via Coresets

Algorithm Distributed k-center

Require: Set of points X_i in machine i for $1 \leq i \leq m$, $\epsilon > 0$

1: Each machine i computes an ϵ-coreset C_i of X_i in round 1

2: Each machine i except 1 sends C_i to machine 1 in round 2

3: Machine 1 computes a k-center clustering of $\cup_i C_i$ and sends the k centers to all machines in round 3

Space complexity: $O(mk(d/\epsilon)^d)$

Approximation via Coresets

Algorithm Distributed k-center

Require: Set of points X_i in machine i for $1 \leq i \leq m$, $\epsilon > 0$

1: Each machine i computes an ϵ-coreset C_i of X_i in round 1
2: Each machine i except 1 sends C_i to machine 1 in round 2
3: Machine 1 computes a k-center clustering of $\cup_i C_i$ and sends the k centers to all machines in round 3

Space complexity: $O(mk(d/\epsilon)^d)$

We show: the union $C = \cup_i C_i$ is an ϵ-coreset of X

Approximation via Coresets

Algorithm Distributed k-center

Require: Set of points X_i in machine i for $1 \leq i \leq m$, $\epsilon > 0$

1: Each machine i computes an ϵ-coreset C_i of X_i in round 1
2: Each machine i except 1 sends C_i to machine 1 in round 2
3: Machine 1 computes a k-center clustering of $\cup_i C_i$ and sends the k centers to all machines in round 3

Space complexity: $O(mk(d/\epsilon)^d)$

We show: the union $C = \cup_i C_i$ is an ϵ-coreset of $X \Longrightarrow$ we obtain a $2(1 + \epsilon)$-approximation

Additive Property of Coresets

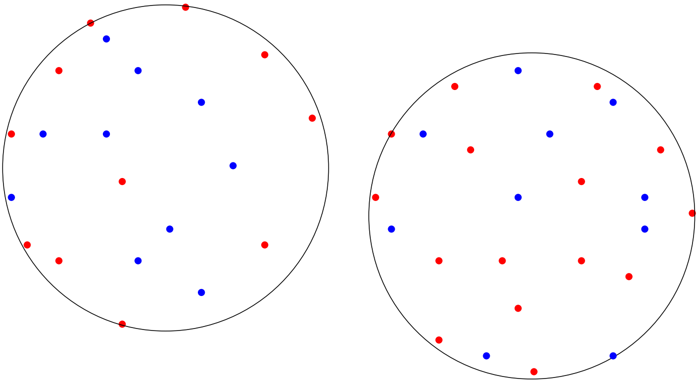
If C_1 is an ϵ-coreset of X_1 and C_2 is an ϵ-coreset of X_2, then $C_1 \cup C_2$ is an ϵ-coreset of $X_1 \cup X_2$.

Additive Property of Coresets

If C_1 is an ϵ-coreset of X_1 and C_2 is an ϵ-coreset of X_2, then $C_1 \cup C_2$ is an ϵ-coreset of $X_1 \cup X_2$.

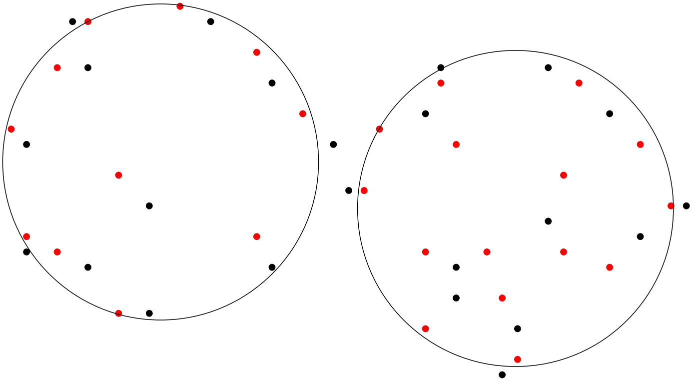
- Consider any clustering T of $C_1 \cup C_2$
- Need to show: the ϵ-expansion of T contains $X_1 \cup X_2$

Clustering T of $C_1 \cup C_2$



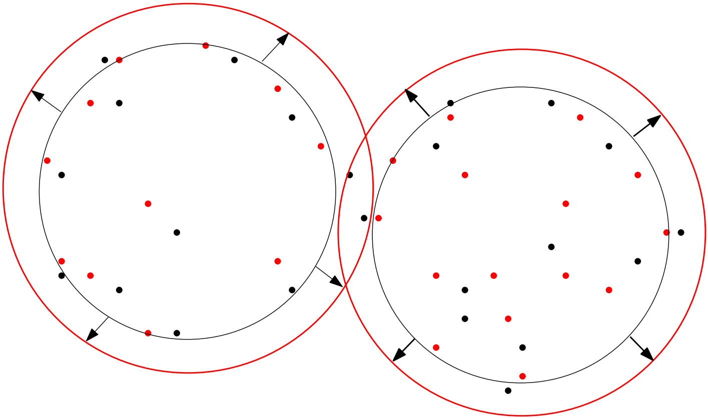
C_1: Red, C_2: Blue

Focus on C_1



The clusters in T might not cover all points of X_1

ϵ-expansion of T contains X_1



Additive Property of Coresets

If C_1 is an ϵ-coreset of X_1 and C_2 is an ϵ-coreset of X_2, then $C_1 \cup C_2$ is an ϵ-coreset of $X_1 \cup X_2$.

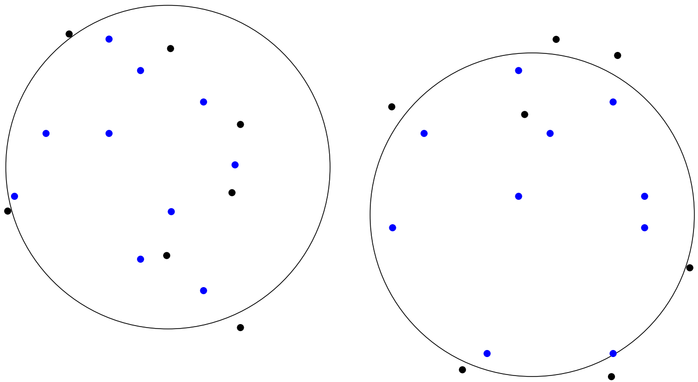
- Consider any clustering T of $C_1 \cup C_2$
- Need to show: the ϵ-expansion of T contains $X_1 \cup X_2$
- ϵ-expansion of T contains X_1

Additive Property of Coresets

If C_1 is an ϵ-coreset of X_1 and C_2 is an ϵ-coreset of X_2, then $C_1 \cup C_2$ is an ϵ-coreset of $X_1 \cup X_2$.

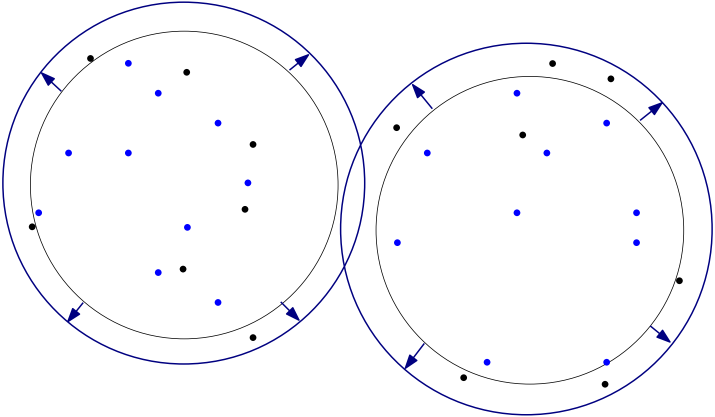
- Consider any clustering T of $C_1 \cup C_2$
- Need to show: the ϵ-expansion of T contains $X_1 \cup X_2$
- ϵ-expansion of T contains X_1
- Similarly, ϵ-expansion of T contains X_2

Focus on C_2



The clusters in T might not cover all points of X_2

ϵ-expansion of T contains X_2



Approximation via Coresets

Algorithm Distributed k-center

Require: Set of points X_i in machine i for $1 \leq i \leq m$, $\epsilon > 0$

1: Each machine i computes an ϵ-coreset C_i of X_i in round 1
2: Each machine i except 1 sends C_i to machine 1 in round 2
3: Machine 1 computes a k-center clustering of $\cup_i C_i$ and sends the k centers to all machines in round 3

Space complexity: $O(mk(d/\epsilon)^d)$

We show: the union $C = \cup_i C_i$ is an ϵ-coreset of $X \Rightarrow$ we obtain a $2(1+\epsilon)$-approximation
Proved!

Outline

The Setting

- n data points arrive one at a time

The Setting

- n data points arrive one at a time
- The goal is to compute k centers of an approximate clustering at the end

The Setting

- n data points arrive one at a time
- The goal is to compute k centers of an approximate clustering at the end
- We have only sub-linear space

The Setting

- n data points arrive one at a time
- The goal is to compute k centers of an approximate clustering at the end
- We have only sub-linear space

The Setting

- n data points arrive one at a time
- The goal is to compute k centers of an approximate clustering at the end
- We have only sub-linear space

We will maintain an ϵ-coreset of the arrived points at each step

The Algorithm

Algorithm Streaming k-center

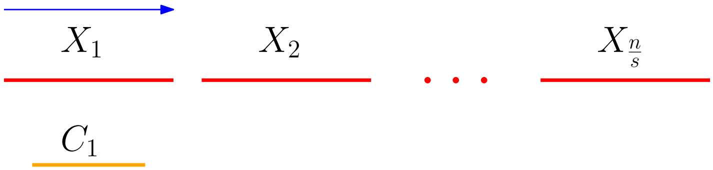
Require: Data stream X of n points, an integer s (bucket size)

$i \leftarrow 1$
repeat
    Add arriving data point p to X_i
    if $|X_i| == s$ **then**
        Compute an ϵ-coreset C_i of X_i
        Remove X_i
        $i \leftarrow i + 1$
    end if
until there are no points left
return $\cup_i C_i$

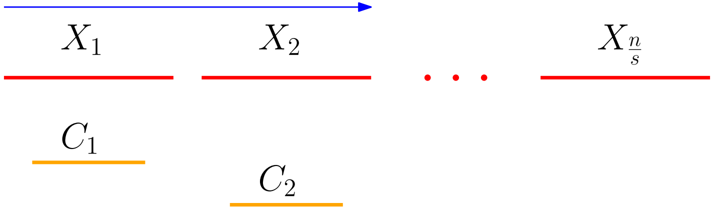
An Example

$$X_1 \qquad X_2 \qquad \cdots \qquad X_{\frac{n}{s}}$$

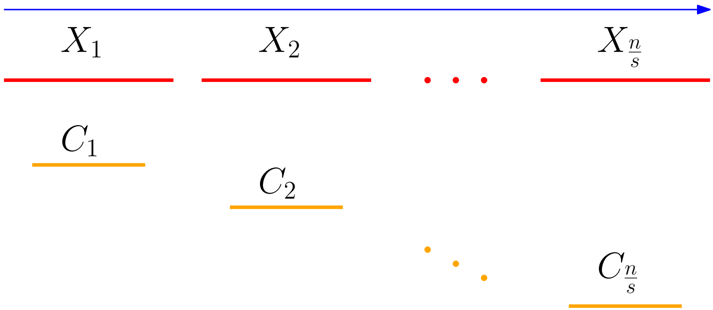
An Example



An Example



An Example



The Analysis

Algorithm Streaming k-center

Require: Data stream X of n points, an integer s (bucket size)

$i \leftarrow 1$
repeat
    Add arriving data point p to X_i
    if $|X_i| == s$ **then**
        Compute an ϵ-coreset C_i of X_i
        Remove X_i
        $i \leftarrow i + 1$
    end if
until there are no points left
return $\cup_i C_i$

The Analysis

Algorithm Streaming k-center

Require: Data stream X of n points, an integer s (bucket size)

$i \leftarrow 1$
repeat
    Add arriving data point p to X_i
    if $|X_i| == s$ **then**
        Compute an ϵ-coreset C_i of X_i
        Remove X_i
        $i \leftarrow i + 1$
    end if
until there are no points left
return $\cup_i C_i$

$\cup_i C_i$ is an ϵ-coreset of X due to the additive property

The Analysis

Algorithm Streaming k-center

Require: Data stream X of n points, an integer s (bucket size)

$i \leftarrow 1$
repeat
    Add arriving data point p to X_i
    if $|X_i| == s$ **then**
        Compute an ϵ-coreset C_i of X_i
        Remove X_i
        $i \leftarrow i + 1$
    end if
until there are no points left
return $\cup_i C_i$

$\cup_i C_i$ is an ϵ-coreset of X due to the additive property

Space complexity: $O(s + \frac{n}{s} \cdot k(d/\epsilon)^d)$

The Analysis

Algorithm Streaming k-center

Require: Data stream X of n points, an integer s (bucket size)

```
i ← 1
repeat
    Add arriving data point p to X_i
    if |X_i| == s then
        Compute an ε-coreset C_i of X_i
        Remove X_i
        i ← i + 1
    end if
until there are no points left
return ∪_i C_i
```

$\cup_i C_i$ is an ϵ-coreset of X due to the additive property

Space complexity: $O(s + \frac{n}{s} \cdot k(d/\epsilon)^d) = O(\sqrt{nk(d/\epsilon)^d})$ setting $s = \sqrt{nk(d/\epsilon)^d}$