MapReduce and Streaming Algorithms for Center-Based Clustering in Doubling Spaces

Geppino Pucci

DEI, University of Padova, Italy

Based on joint works with:

M. Ceccarello, A. Mazzetto, and A. Pietracaprina

Center-based clustering

Center-based clustering in general metric spaces: Given a pointset S in a metric space with distance $d(\cdot,\cdot)$, determine a set $C^\star \subseteq S$ of k centers minimizing:

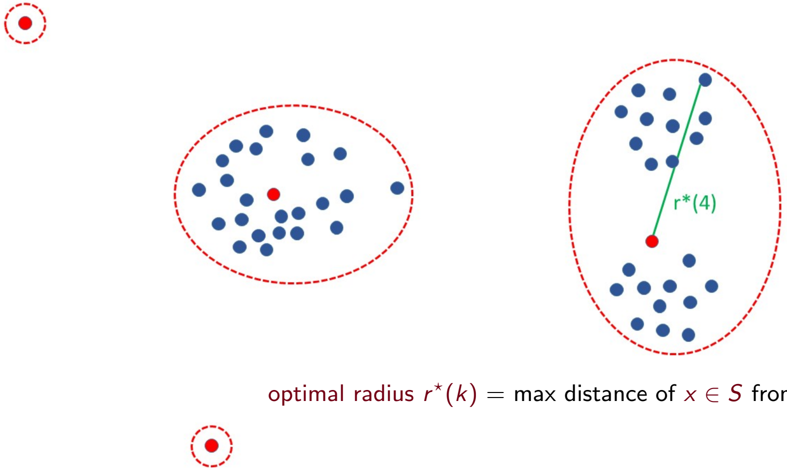
- $\max_{p \in S} \{d(p, C^\star)\}$ (k-**center**)
- $\sum_{p \in S} d(p, C^\star)$ (k-**median**)
- $\sum_{p \in S} (d(p, C^\star))^2$ (k-**means**)

Remark: On general metric spaces it makes sense to require that $C^\star \subseteq S$. This assumption is often relaxed in Euclidean spaces (continuos vs discrete version)

Variant: Center-based clustering with z outliers: Disregard the z largest distances when computing the objective function.

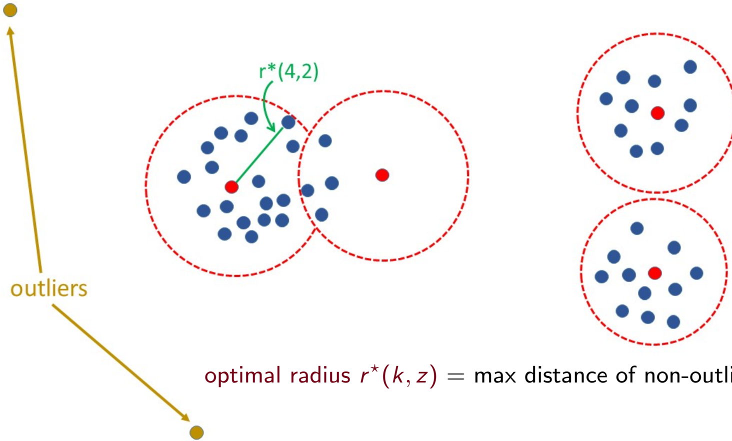
Example: pointset instance

Example: solution to 4-center



optimal radius $r^\star(k)$ = max distance of $x \in S$ from $C^\star$

Example: solution to 4-center with 2 outliers



optimal radius $r^\star(k,z)$ = max distance of non-outlier $x \in S$

Center-based clustering for big data

1. Deal with very large pointsets
   - MapReduce distributed setting
   - Streaming setting
2. **Aim:** try to match best sequential approximation ratios with limited local/working space
3. Very simple algorithms with good practical performance
4. Concentrate on k-center with and without outliers [CeccarelloPietracaprinaP, VLDB2019].
5. End of the talk: sketch very recent results for k-median and k-means [MazzettoPietracaprinaP, arXiv 2019]

Outline

- Background
  - MapReduce and Streaming models
  - Previous work
  - Doubling Dimension
- k center (with and without outliers):
  - Summary of results
  - Coreset selection: main idea
  - MapReduce algorithms
  - Porting to the Streaming setting
  - Experiments
- Sketch of new results for k-median and k-means

Background: MapReduce and Streaming models

MapReduce

- Targets distributed cluster-based architectures
- Computation: sequence of *rounds* where data (*key-value* pairs) are mapped by key into subsets and processed in parallel by reducers equipped with small local memory
- Goals: few rounds, (substantially) sublinear local memory, linear aggregate memory.

Streaming

- Data provided as a continuous stream and processed using small working memory
- Multiple passes over data may be allowed
- Goals: 1 (or few) pass(es), (substantially) sublinear working memory

Background: Previous work

- Sequential algorithms for general metric spaces:
  - k-center: 2-approximation ($O(nk)$ time) and $2-\epsilon$ inapproximability [Gonzalez85]
  - k-center with z outliers: 3-approximation ($O(n^2 k \log n)$ time) [Charikar+01]
- MapReduce algorithms:

Reference	Rounds	Approx.	Local Memory
k-center problem			
[Ene+11] (w.h.p.)	$O(1/\epsilon)$	10	$O(k^2\|S\|^\epsilon)$
[Malkomes+15]	2	4	$O((\|S\|k)^{1/2})$
k-center problem with z outliers			
[Malkomes+15]	2	13	$O((\|S\|(k+z))^{1/2})$

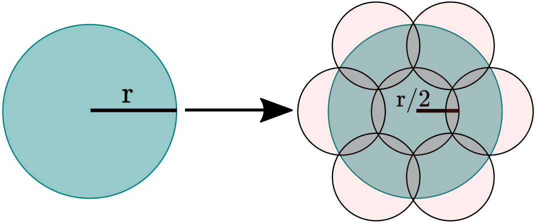
Background: Previous work (cont'd)

- Streaming algorithms:

Reference	Passes	Approx.	Working Memory
k-center problem			
[McCutchen+08]	1	$2+\epsilon$	$O\left(k\epsilon^{-1}\log\epsilon^{-1}\right)$
k-center problem with z outliers			
[McCutchen+08]	1	$4+\epsilon$	$O\left(kz\epsilon^{-1}\right)$

Background: doubling dimension

Our algorithms are analyzed in terms of the doubling dimension D of the metric space: $\forall r$: any *ball* of radius r is covered by $\leq 2^D$ balls of radius $r/2$



- Euclidean spaces
- Shortest-path distances of mildly expanding topologies
- Low-dimensional pointsets of high-dimensional spaces

Our Algorithms

Model	Rnd/Pass	Approx.	Local/Working Memory
	k-center problem		
MapReduce	2	$2+\epsilon$ (4)	$O\left(\sqrt{\lvert S\rvert k}\,(\frac{4}{\epsilon})^D\right)$ $\left(O\left(\sqrt{\lvert S\rvert k}\right)\right)$
	k-center problem with z outliers		
MapReduce	2	$3+\epsilon$ (13)	$O\left(\sqrt{\lvert S\rvert(k+z)}\,(\frac{24}{\epsilon})^D\right)$ $\left(O\left(\sqrt{\lvert S\rvert(k+z)}\right)\right)$
MapReduce (w.h.p.)	2	$3+\epsilon$	$O\left(\left(\sqrt{\lvert S\rvert(k+\log\lvert S\rvert)}+z\right)(\frac{24}{\epsilon})^D\right)$
Streaming	1	$3+\epsilon$ $(4+\epsilon)$	$(k+z)\,(\frac{96}{\epsilon})^D$ $\left(O\left(\frac{kz}{\epsilon}\right)\right)$

- Substantial improvement in approximation quality at the expense of larger memory requirements (constant factor for constant ϵ, D)
- MR algorithms are oblivious to D
- Large constants due to the analysis. Experiments show practicality of our approach.

Summary of results (cont'd)

Main features

- (Composable) coreset approach: select small $T \subseteq S$ containing good solution for S and then run (adaptation of) best sequential approximation on T
- Flexibility: coreset construction can be either distributed (MapReduce) or streamlined (Streaming)
- Adaptivity: Memory/approximation tradeoffs expressed in terms of the doubling dimension d of the pointset
- Quality: MR and Streaming algorithms using small memory and *almost matching* best sequential approximations.

Coreset selection: main idea

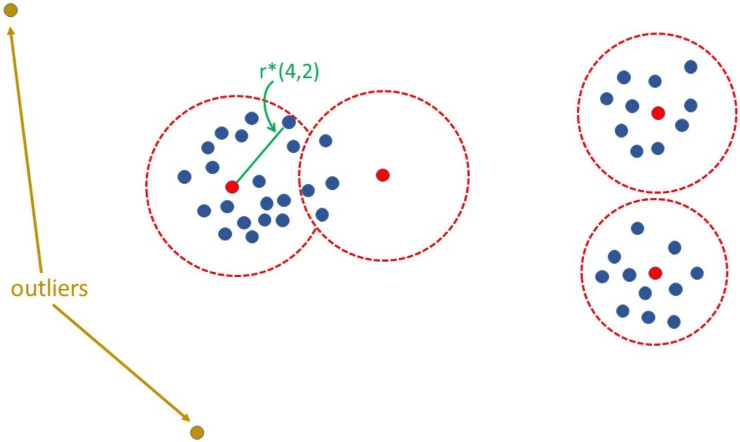
- Let $r^\star$ = max distance of any (non-outlier) $x \in S$ from closest optimal center
- Select a coreset $T \subseteq S$ ensuring that

$$d(x, T) \leq \epsilon r^\star \quad \forall x \in S - T$$

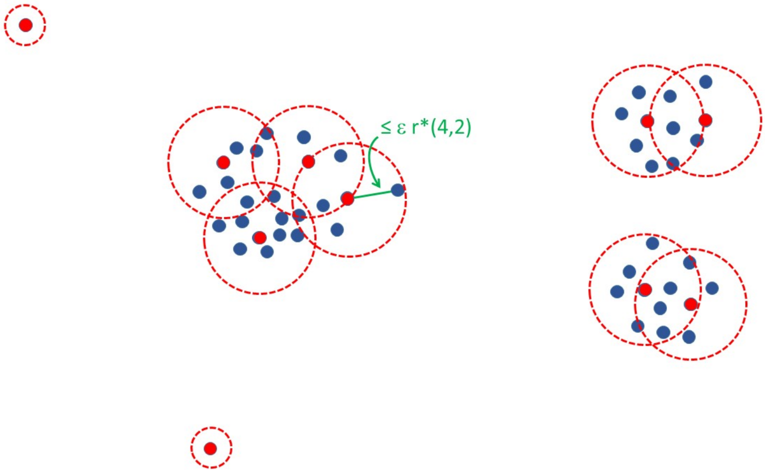
  using sequential h-center approximation, for h suitably larger than k. (Similar idea in [CeccarelloPietracaprinaPUpfal17] for diversity maximization → next talk)
- **Obs:** in general, T must contain outliers

Example: pointset instance

Example: optimal solution k=4, z=2



Example: 10-point coreset T (red points)



Basic primitive for coreset selection (based on [Gonzalez85])

Select(S', h, ϵ):

Input: Subset $S' \subseteq S$, parameters $h, \epsilon > 0$
Output: Coreset $T \subseteq S'$ of size $\geq h$

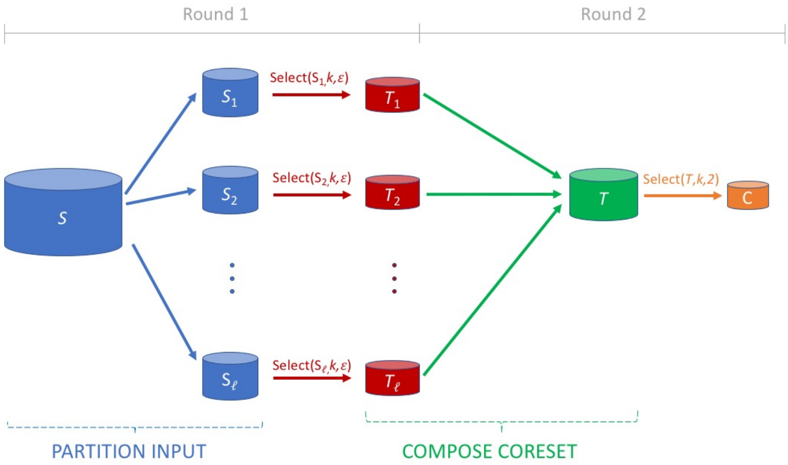
$T \leftarrow$ arbitrary point $c_1 \in S'$
$r(1) \leftarrow$ max distance of any $x \in S'$ from T
for $i = 2, 3, \ldots$ **do**
  Find farthest point $c_i \in S'$ from T, and add it to T
  $r(i) \leftarrow$ max distance of any $x \in S'$ from T
  if $((i \geq h)$ AND $(r(i) \leq (\epsilon/2) r(h)))$ **then return** T

Lemma: Let $r^*(h)$ be the optimal h-center radius for the *entire set* S and let *last* the index of the last iteration of Select. Then:

$$r(last) \leq \epsilon r^*(h)$$

Proof idea: by a simple adaptation of Gonzalez's proof, $r(i = h) \leq 2r^*(h)$

MapReduce algorithms: k-center



MapReduce algorithms: k-center (cont'd)

Analysis

- **Approximation quality:** let $C = \{c_1, \ldots, c_k\}$ be the returned centers.

  For any $x \in S_j$ (arbitrary j)

$$\begin{aligned} d(x, C) &\leq d(x, t) + d(t, C) \quad (t \in T_j \textit{ closest to } x) \\ &\leq \epsilon r^\star(k) + 2r^\star(k) = (2+\epsilon)r^\star(k) \end{aligned}$$

- **Memory requirements:** assume doubling dimension D
  - set $\ell = \sqrt{|S|/k}$
  - Technical lemma: $|T_j| \leq k(4/\epsilon)^D$, for every $1 \leq j \leq \ell$

$\Longrightarrow$ Local memory $= O\left(\sqrt{|S|k}(4/\epsilon)^D\right)$.

Remarks:

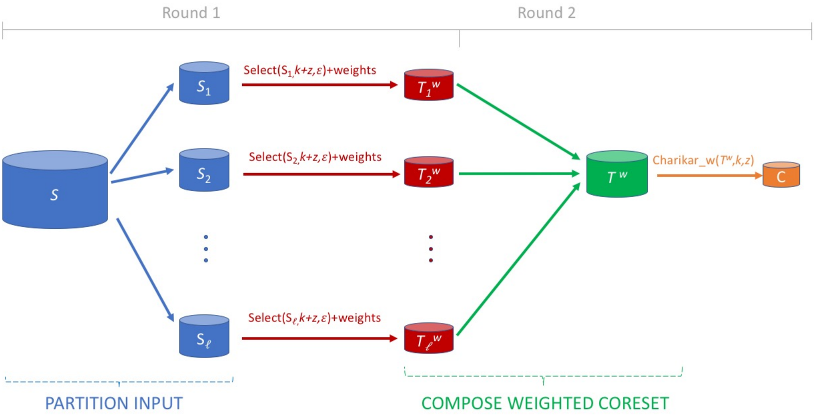
- For constant ϵ and D: $(2+\epsilon)$-approximation with the same memory requirements as the 4-approximation in [Malkomes+15]
- Our algorithm is oblivious to D

MapReduce algorithms: k-center with z outliers

Similar approach to the case without outliers but with some important differences

1. Each coreset $T_j \subseteq S_j$ must contain $\geq k + z$ points (making room for outliers)
2. Each $t \in T_j$ has a weight $w(t)$ = number of points of $S_j - T_j$ for which t is proxy (i.e., closest). Let T_j^w denote the set T_j with weights.
3. On $T^w = \bigcup_j T_j^w$ a suitable weighted variant of the algorithm in [Charikar01+] (dubbed Charikar_w) is run which:
   - determines k suitable centers (final solution) covering most points of T^w
   - uncovered points of T^w have aggregate weight z and are the proxies of the outliers

MapReduce algorithms: k-center with z outliers (cont'd)



MapReduce algorithms: k-center with z outliers (cont'd)

Analysis

- **Approximation quality:** let $C = \{c_1, \ldots, c_k\}$ be the returned centers.

  For any *non-outlier* $x \in S_j$ (arbitrary j) with proxy $t \in T_j^w$

  $$d(x,t) \leq \epsilon r^\star(k,z) \text{ and } d(t,C) \leq (3+5\epsilon)r^\star(k,z)$$

  $\Longrightarrow$ $(3+\epsilon')$-approximation for every $\epsilon' > 0$.
- **Memory requirements:** assume doubling dimension D
  - set $\ell = \sqrt{|S|/(k+z)}$
  - Technical lemma: $|T_j| \leq (k+z)(4/\epsilon)^D$, for every $1 \leq j \leq \ell$

  $\Longrightarrow$ Local memory $= O\left(\sqrt{|S|(k+z)}(4/\epsilon)^D\right)$.

Remarks:

- For constant ϵ and D: $(3+\epsilon)$-approximation with the same memory requirements as the 13-approximation in [Malkomes+15]
- Our algorithm is oblivious to D

MapReduce algorithms: k-center with z outliers (cont'd)

Randomized Variant

- Create $S_1, S_2, \ldots, S_\ell$ as a random partition

  ($\Rightarrow z' = O(z/\ell + \log|S|)$ outliers per partition *w.h.p.*)
- Execute the deterministic algorithm with z' in lieu of z

Analysis

- **Approximation quality:** $(3 + \epsilon')$ (as before)
- **Memory requirements:** $O\left(\left(\sqrt{|S|(k + \log|S|)} + z\right)(24/\epsilon)^D\right)$

Remark:

- For constant ϵ and D: $O\left(\sqrt{|S|(k + \log|S|)} + z\right)$ local memory (linear dependence on z desirable)

Streaming algorithm: k-center with z outliers

Main idea: single-pass simulation of MR-algorithm with no data partition ($\ell = 1$)

Algorithm:

- Obtain coreset T^w by running a weighted variant of the doubling algorithm of [Charikar+04] for τ-center (without outliers), with $\tau = (k+z)(96/\epsilon)^D$ on the input stream.

  Remark: D must be known! (obliviousness with 1 extra pass)

- Run Charikar_w in working memory on T^w to obtain final solution.

Analysis: reasoning as for the MR-algorithm we can prove

- $(3+\epsilon')$-approximation for every $\epsilon' > 0$.
- Working memory $= O\left((k+z)(96/\epsilon)^D\right)$

Experiments

Goals of the experiments:

1. Assess quality of solution as a function of the coreset size
2. Assess scalability of the MR-algorithms

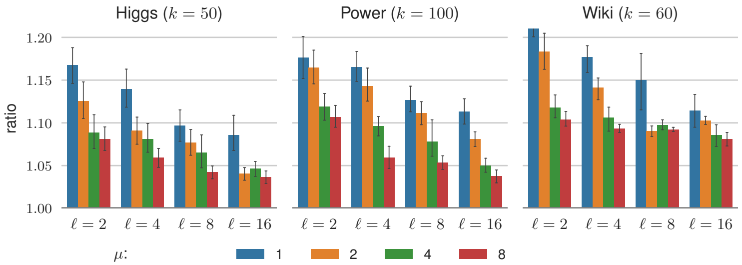
Datasets:

- Higgs: $\simeq$ 11M points (in $\Re^7$) from high-energy physics experiments
- Power: $\simeq$ 2M points (in $\Re^7$) from electric power consumption measurements
- Wiki: $\simeq$ 5M pages ($\stackrel{\texttt{word2vec}}{\rightarrow}$ vectors in $\Re^{50}$)
- Inflated instances of Higgs/Power/Wiki: up to 100 times larger (for scalability)

Platform: Cluster with

- 16 4-core I7 processor with 18GB RAM
- 10Gb Ethernet

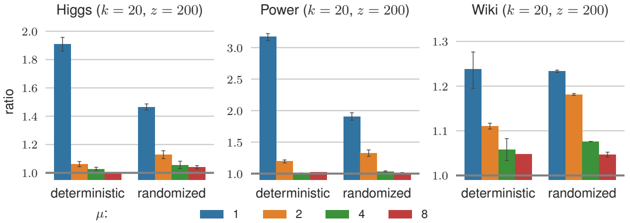
Accuracy vs coreset size/parallelism: *k*-center



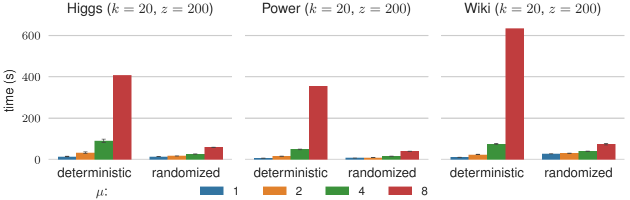
Approx. ratio vs. coreset size $\mu \cdot k$, with $\mu = 1, 2, 4, 8$ and $\ell = 2, 4, 8, 16$

Remark: Approximation ratio measured against best solution ever computed for the specific instance (max parallelism, max memory)

Accuracy vs coreset size/parallelism: k-center with outliers

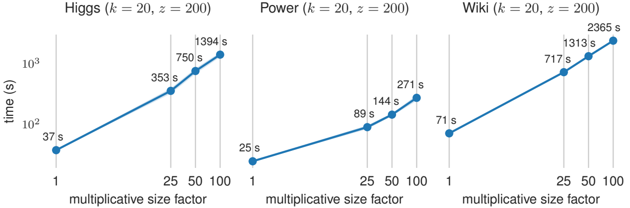


Approx. ratio vs. coreset size $\mu \cdot k$, with $\mu = 1, 2, 4, 8$, $\ell = 16$ (det/rand)

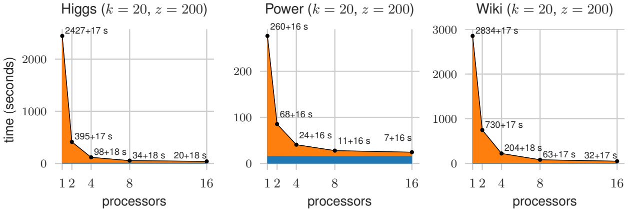


Running times (same parameters)

Scalability: *k*-center with outliers



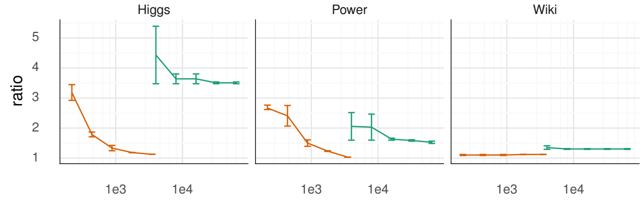
Running time vs input size (randomized, fixed parallelism $\ell = 16$)



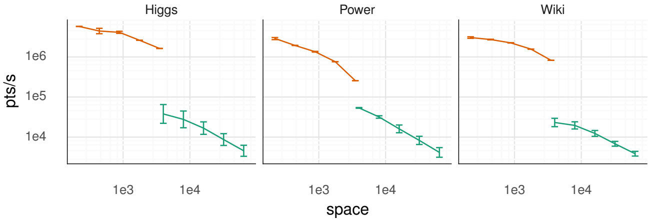
Running time vs # processors (randomized, fixed final coreset size)

Orange area: coreset construction. Blue area: seq. solution on coreset

Streaming performance: *k*-center with outliers



Accuracy vs working space: Ours (orange) vs [McCutchen+08] (green)



Throughput (pts/s) vs working space

Recent developments for k-median and k-means

- Composable coreset constructions for both problems for general metric spaces (centers belong to S)
- Results: 2-round MapReduce algorithms for both problems:
  - Approximation ratio: $\alpha + \epsilon$, α best sequential approximation for the problem, $\epsilon \in (0, 1)$
  - Local space: $\tilde{O}\left(\sqrt{|S|k}(c/\epsilon)^{2D}\right)$. Sublinear for $d = O(1)$.
- First distributed algorithms for general spaces to achieve (almost) sequential accuracy
- **Main Idea:** Obtain each local coreset T_i as the centers of a ball decomposition of S_i aimed at refining initial (bicriteria) constant approximation for S_i (inspired by exponential grids of [Har-Peled+04-05] for $\Re^d$).
- Simple, deterministic construction
- Check `arxiv.org/abs/1904.12728` for more

Conclusions

- (Composable) coreset constructions for k-center (w/o and with z outliers), k-median, k-means
- Coresets enable a spectrum of space/quality tradeoffs
- Approximation guarantees for MR and Streaming can get arbitrarily close to best sequential ones
- Experimental evidence of practicality of approach (k-center)

Future Work

- Smaller coresets (non uniform sampling?)
- Streaming algorithms and experiments for k-median and k-means