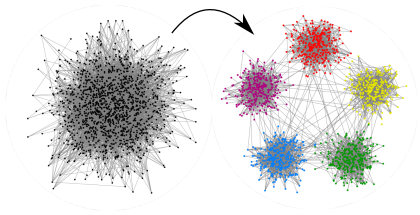
Sublinear Algorithms for Euclidean Clustering and Correlation Clustering

Vincent Cohen-Addad

Google Research

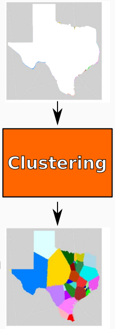


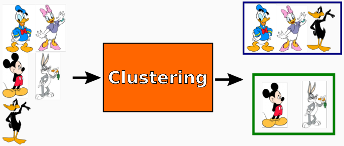
Clustering: A Classic Data Analysis Task

Partition data points according to *similarity*:
Similar points should be in the same part.

Geographic distances

E.g.:
Redistricting,
Facility location





Distances represent similarities

E.g.:
similarities between data elements
(images, texts, musics, DNA, etc.)

This talk:

Clustering of:

- Euclidean metrics
- Graphs

Euclidean Clustering:

(k, z)-Clustering:

- Input: A *point set* $X \subset \mathbb{R}^d$;
- Output: A set of k representatives $C \subset \mathbb{R}^d$, called *centers* s.t.:
  - $|C| = k$
  - That minimizes $\sum_{x \in X} \min_{c \in C} ||c - x||_p^z$

This talk $p = 2$, we work with Euclidean distances.

k-median $\iff z = 1$
k-means $\iff z = 2$

Sublinear Algorithms?

Observation: Solution size is k points in $\mathbb{R}^d$
(so we are ok with a running time that is polynomial in kd)

Sublinear Algorithms?

Observation: Solution size is k points in $\mathbb{R}^d$
(so we are ok with a running time that is polynomial in kd)

Observation: If no good solution with balanced clusters $\implies$ needle in a haystack phenomenon.

Sublinear Algorithms?

Observation: Solution size is k points in $\mathbb{R}^d$
(so we are ok with a running time that is polynomial in kd)

Observation: If no good solution with balanced clusters $\implies$ needle in a haystack phenomenon.

Assumption: OPT clusters are of balanced size and we look for a running time of $\Theta(kd)$.

Sublinear Algorithms?

Observation: Solution size is k points in $\mathbb{R}^d$
(so we are ok with a running time that is polynomial in kd)

Observation: If no good solution with balanced clusters $\implies$ needle in a haystack phenomenon.

Assumption: OPT clusters are of balanced size and we look for a running time of $\Theta(kd)$.

This Talk: Approximate $(1, z)$-clustering with few samples

Power Mean Objective

$(1, z)$-clustering is also known as *Power mean objective*

Power Mean Objective

$(1, z)$-clustering is also known as *Power mean objective*

Our question:

How many points from the input are needed to find an $(1+\varepsilon)$-approximation to the power mean of the whole input?

Why do we care about $z \notin \{1, 2\}$?

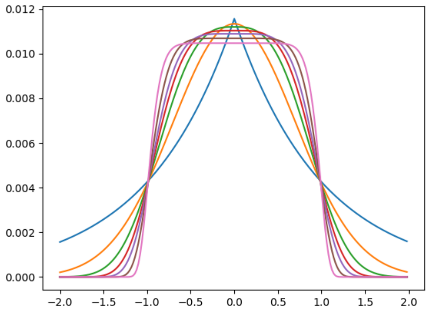
Why Power Mean?

- Max-likelihood estimator of a Generalized normal distribution $\sim \exp(-|x - \mu|^z)$

Why do we care about $z \notin \{1, 2\}$?

Why Power Mean?

- Max-likelihood estimator of a Generalized normal distribution $\sim \exp(-|x - \mu|^z)$
- Taking a larger z approximates the Minimum Enclosing Ball objective (find the smallest ball containing the input)



Why do we care about $z \notin \{1, 2\}$?

Why Power Mean?

- Max-likelihood estimator of a Generalized normal distribution $\sim \exp(-|x-\mu|^z)$
- Taking a larger z approximates the Minimum Enclosing Ball objective (find the smallest ball containing the input)

Approach: Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for *all* sets S of k centers it holds

$$|\mathrm{cost}_w(\Omega, S) - \mathrm{cost}(A, S)| \leq \varepsilon \cdot \mathrm{cost}(A, S)$$

Approach: Coreset

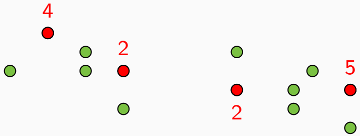
Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for *all* sets S of k centers it holds

$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$

Approach: Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for *all* sets S of k centers it holds

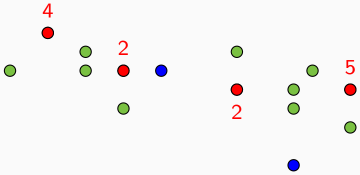
$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$



Approach: Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for *all* sets S of k centers it holds

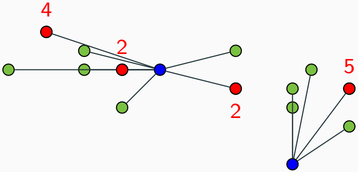
$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$



Approach: Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for *all* sets S of k centers it holds

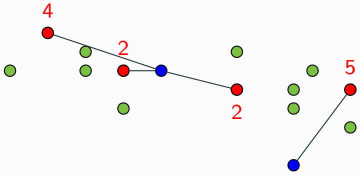
$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$



Approach: Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for *all* sets S of k centers it holds

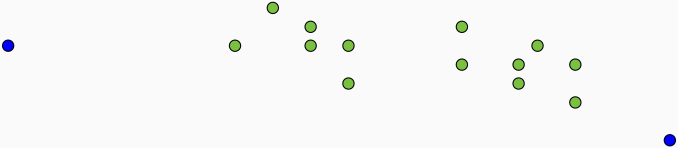
$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$



Approach: Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for *all* sets S of k centers it holds

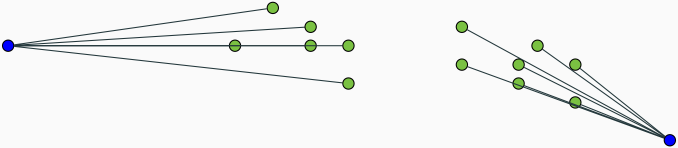
$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$



Approach: Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for *all* sets S of k centers it holds

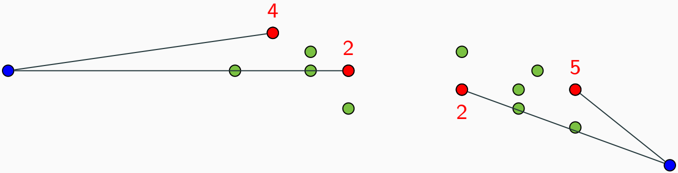
$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$



Approach: Coreset

Given a set of points A, a weighted subset $\Omega \subset A$ is a (k, ε)-coreset if for *all* sets S of k centers it holds

$$|\text{cost}_w(\Omega, S) - \text{cost}(A, S)| \leq \varepsilon \cdot \text{cost}(A, S)$$



Coreset for Sublinear Time Algorithms

Weak Coresets

Given a point set A, a weighted point set Ω is a weak (k, ε)-coreset if for any point c' such that $\text{cost}_w(\Omega, c') \leq (1+\varepsilon) \min_{c \in \mathbb{R}^d} \text{cost}_w(\Omega, c)$ we have

$$\text{cost}(A, c') \leq (1 + O(\varepsilon)) \min_{c \in \mathbb{R}^d} \text{cost}(A, c)$$

Our Contribution: Weak Coreset Constructed by Uniform Sampling

State of the Art

Weak coresets of size $\tilde{O}(\varepsilon^{-2} \cdot \min(\varepsilon^{-2}, d))$ (see e.g. [Feldman, Langberg, STOC' 11]).

Theorem – C.-A., Saulpic, Schwiegelshohn'21

One can construct a weak coreset of size $2^{O(z)}\varepsilon^{-2}$ by sampling $\tilde{O}(\varepsilon^{-z-3})$ points.

To obtain a $(1+\varepsilon)$-approximation algorithm with constant probability, one need to query at least $\Omega(\varepsilon^{-z+1})$ points, even when $d = 1$.

Naive Approach and Analysis

Algorithm:

Sample δ points uniformly at random. Assign weight n/δ.

Analysis for a fixed center s:

In Expectation: $\mathbb{E}[\text{cost}_w(\Omega, s)] = \text{cost}(A, s)$

$$\begin{aligned}\mathbb{E}[\text{cost}_w(\Omega, s)] &= \mathbb{E}\left[\sum_{p \in \Omega} \frac{n}{\delta} \cdot ||p - s||_2^z\right] \\ &= \sum_{p \in A} \frac{n}{\delta} \cdot ||p - s||_2^z \cdot \Pr[p \in \Omega] \\ &= \sum_{p \in A} ||p - s||_2^z = \text{cost}(A, s)\end{aligned}$$

For a fixed center s, we are happy!

Naive Approach and Analysis

Algorithm:

Sample δ points uniformly at random. Assign weight n/δ.

Challenge

We would like to have this holds **for all near-optimal** s simultaneously.
$\iff$ We look for concentration bounds.

Variance Reduction

Observation:

If all the points contribute the same amount to the objective,
Then good concentration using e.g.: Hoeffding inequality.

Idea

1. Partition the points into groups s.t.: points in the same group contribute the same amount to the objective.
2. Apply uniform sampling within the groups.

Idea

1. Partition the points into groups s.t.: points in the same group contribute the same amount to the objective.
   Not very well defined: contribution of a point depends on the location of the center!

Intuition: Points that contributes the same amount in an approximate solution S are not too far from each other.

$\iff$ we can tolerate an error proportional to ε times their contribution in S.

Idea

1. Partition the points into groups s.t.: points in the same group contribute the same amount to the objective.
   Not very well defined: contribution of a point depends on the location of the center!
   Fix: points in the same group contribute the same amount in an approximate solution

Intuition: Points that contributes the same amount in an approximate solution S are not too far from each other.

$\iff$ we can tolerate an error proportional to ε times their contribution in S.

Algorithm and Analysis

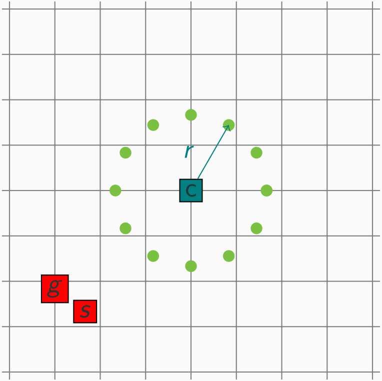
1. Sample a point q u.a.r.
   a good approximation
2. Sample a set S of $\tilde{O}(\varepsilon^{-z-3})$ points u.a.r.
3. Compute the maximum distance ℓ such that there exist $\approx 2/3\varepsilon^{z+1}|S|$ points with distance at least d from q.
   Discard all points at distance greater than d.
   "Variance reduction": Remove far points that have high contribution to the cost.
4. Define groups R_i s.t. $R_i \cap S$ contains all the points at distance $(d \cdot 2^{-i}, d \cdot 2^{-i+1}]$ from q.
5. For all i s.t. $|R_i \cap S| \leq\approx \varepsilon^{z+1}|S|$, remove all points in $R_i \cap S$ from S.
   Remaining points form the coreset.
6. Solve the problem on the coreset S.

Main Arguments

- Infinitely many solutions s!

Main Arguments

- Problem is intrinsically **low-dimensional** because we look for one center. $\exists$ Discretization of $\mathbb{R}^d \implies$ small number of $(1+\varepsilon)$-approx solutions that are different.



Main Arguments

Small number of "interesting solutions"

Combined with

Chaining: Inductive analysis showing that as we sample more and more points the error gets smaller and smaller.

Recent for Euclidean space

Feldman, Langberg (STOC11)	$O(dk \log k\epsilon^{-2z})$
$*$ Sohler, Woodruff (FOCS18)	$O((k/\varepsilon)^{O(z)})$
Huang, Vishnoi (STOC20)	$O(k \log^2 k\epsilon^{-2-2z})$
Braverman, Jiang, Krauthgamer, Wu (SODA21)	$O(k^2 \log^2 k\epsilon^{-4})$
C.-A., Saulpic, Schwiegelshohn (STOC21)	$\tilde{O}(k\epsilon^{-2-\max(2,z)})$
C.-A., Saulpic, Schwiegelshohn (Neurips21)	$O(2^z\varepsilon^{-2})$
C.-A., Larsen, Saulpic., Schwiegelshohn (STOC22)	$\tilde{O}(k\epsilon^{-2}\min(k2^z, \varepsilon^{-z}))$

The Power of Uniform Sampling

[Braverman, C.-A., Krauthgamer, Jiang, Schwiegelshohn, Toftrup, Xuan FOCS'22]

New framework for uniform sampling $\Longrightarrow$ new bounds for k-clustering with extra constraints capacitated, fair, etc..

Further Recent Results

Finite Metrics

Feldman, Langberg (STOC'11)	$O(k\varepsilon^{-2z} \log n \log k)$
C.-A., Saulpic, Schwiegelshohn	$O(k\varepsilon^{-\max(2,z)} \log n)$

Doubling Metrics of dim. D

Huang, Jiang, Li, Wu (FOCS'18)	$\tilde{O}(k^3 D\varepsilon^{-\max(2,z)})$
C.-A., Saulpic, Schwiegelshohn	$\tilde{O}(kD\varepsilon^{-\max(2,z)})$

Graphs with Treewidth t

Baker, Braverman, Huang, Jiang, Krauthgamer, Wu (ICML'20)	$O(k^3 t\varepsilon^{-2})$
C.-A., Saulpic, Schwiegelshohn	$\tilde{O}(kt\varepsilon^{-\max(2,z)})$

Minor-free Graphs

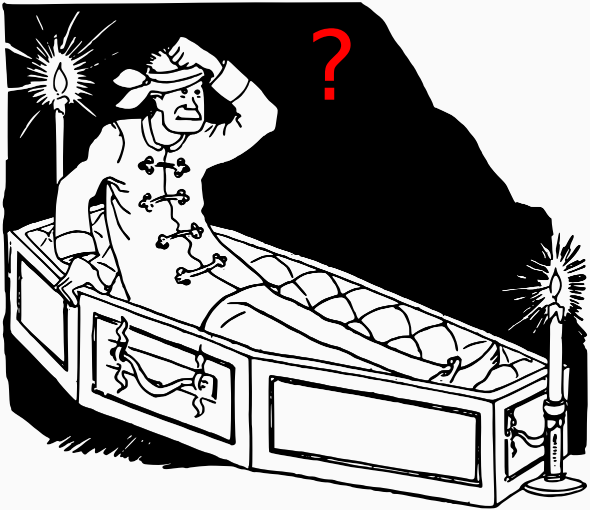
Braverman, Jiang, Krauthgamer, Wu (SODA'21)	$O(k^2\varepsilon^{-4})$
C.-A., Saulpic, Schwiegelshohn	$O(k \log^2 k\varepsilon^{-6})$

Future Challenges

- Closing the gap for Euclidean coreset bounds:
  k-means: $\tilde{O}(k\varepsilon^{-4})$ vs $\Omega(k\varepsilon^{-2})$.

- Coresets for other problems? Set cover, submodular optimization? In statistics?

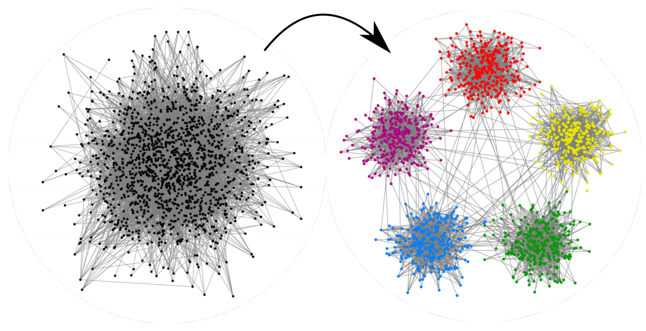
Intermission



Graph Clustering

Similarity is given by edges, two adjacent nodes are similar.
Goal: Identify dense subgraphs

Input: A social network, set of genes of species, the world wide web.



Goal: Find communities in social networks, groups of related organisms, designing

Correlation Clustering:

Input: A complete graph, each edge e has a label $\ell_e \in \{+, -\}$.
Goal: A partition $\{V_1, \ldots, V_k\}$ of V that minimizes

$$\sum_{i=1}^{k} \sum_{u \in V_i} \sum_{v \notin V_i} [\ell_{(u,v)} = +] \ + \ \sum_{u \in V_i} \sum_{v \in V_i} [\ell_{(u,v)} = -]$$

Intuition:

Pay each edge (u, v) where $\ell_{(u,v)} = +$ if u and v are in $\neq$ clusters.
Pay each edge (u, v) where $\ell_{(u,v)} = -$ if u and v are in same cluster.

In practice: $-$-edges are the "no-edges", $+$-edges are "normal edges".

Previous classic work

A simple "pivot-based" 3-approximation by [Ailon, Charikar Newman '04]:
- Pick a random vertex, put it and all its $+$-neighbor in a cluster - Recurse on the rest.

An LP-rounding-based 2.06-approximation by [Chawla, Makarychev, Schramm, Yaroslavtsev '15]:
- Solve the LP
- Round it using a pivot-based approach.

[NEW! C.-A., Lee, Newman '22]

A Sherali-Adams-LP-rounding-based 1.994-approximation.

Why Correlation Clustering

- G consists of disjoint cliques $C_1, \ldots, C_k \implies$ Min Correlation Clustering Cost is 0.
- The number of clusters is function the input

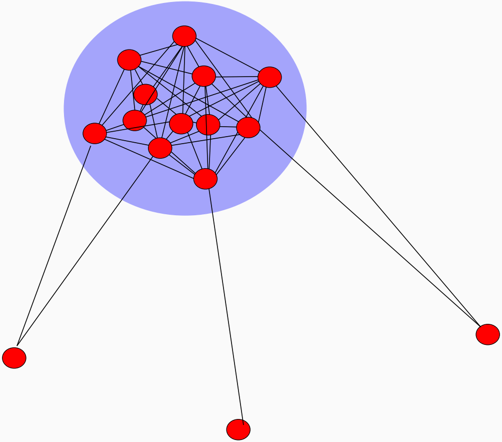
Important Properties

Clusters are very dense $+$-edges subgraphs with little expansion.

There exists an $O(1)$-approx such that:

- Clusters have $+$-edge density $\geq .9$, and
- Each vertex has $\geq .9$ fraction of its $+$-neighbors inside its own cluster.

Clusters we are interested in



Agreement

Key Insight

Symmetric difference between $+$-neighborhood sets of two vertices in the same cluster is small.
If u, v in same cluster, then $|N^+(u) \Delta N^+(v)|$ is much smaller than $\max(|N^+(u)|, |N^+(v)|)$.

Agreement

Key Insight

Symmetric difference between $+$-neighborhood sets of two vertices in the same cluster is small.
If u, v in same cluster, then $|N^+(u) \Delta N^+(v)|$ is much smaller than $\max(|N^+(u)|, |N^+(v)|)$.

Lemma

There exists an $O_\varepsilon(1)$-approximation to correlation clustering such that for any u, v in the same cluster, then

$$|N^+(u) \Delta N^+(v)| \leq \varepsilon \max(|N^+(u)|, |N^+(v)|).$$

Call such pairs of vertices in *agreement*.

Simple Parallel Algorithm

ParallelCorrelationClustering:

1. Discard all $+$-edges (u, v) whenever u and v are not in agreement. We know they are not in the same cluster anyway.

Simple Parallel Algorithm

ParallelCorrelationClustering:

1. Discard all $+$-edges (u, v) whenever u and v are not in agreement. We know they are not in the same cluster anyway.
2. Call a vertex *light* if its $+$-degree has decreased by $\Omega(1)$.
   Discard all $+$-edges between light vertices.
   Vertices of very dense subgraphs with low expansion are not light.

Simple Parallel Algorithm

ParallelCorrelationClustering:

1. Discard all $+$-edges (u, v) whenever u and v are not in agreement. We know they are not in the same cluster anyway.
2. Call a vertex *light* if its $+$-degree has decreased by $\Omega(1)$.
   Discard all $+$-edges between light vertices.
   Vertices of very dense subgraphs with low expansion are not light.
3. Compute the connected components of the resulting graph, these are the correlation clustering clusters.

Simple Parallel Algorithm

ParallelCorrelationClustering:

1. Discard all $+$-edges (u, v) whenever u and v are not in agreement. We know they are not in the same cluster anyway.
2. Call a vertex *light* if its $+$-degree has decreased by $\Omega(1)$.
   Discard all $+$-edges between light vertices.
   Vertices of very dense subgraphs with low expansion are not light.
3. Compute the connected components of the resulting graph, these are the correlation clustering clusters.
   Connected components have diameter at most 4 so can be done efficiently!

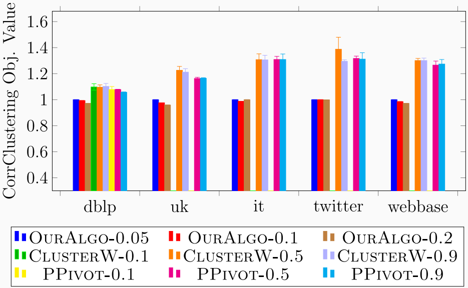
Sampling $O(\log n)$ **neighbors uniformly for each node is enough**

Results: Theory and Practice

[C.-A., Lattanzi, Mitrović, Norouzi-Fard, Parotsidis, Tarnawski '21]

Theorem

MPC-CorrelationClustering achieves an $O(1)$-approximation in $O(1)$ MPC rounds (total memory is $\tilde{O}$(number of $+-$edges)).



Open Problems

- Improved by [Assadi, Wang] and [Behnezhad, Charikar, Ma, Tan] to $3 + \varepsilon$-approximation in $O(1/\varepsilon)$ parallel rounds.
  What is the best approximation one can obtain in time $\tilde{O}(n)$?
  (or $1, 2, 3, 4, \ldots, 10$ rounds in distributed?)
- $O(\log n)$-approximation for the weighted case in time $\tilde{O}(n)$?
- FPT approximation scheme in sublinear time (parameterized by # clusters)?

Future Challenges

- **Lower Bound**: What is the best approximation ratio we can get in sublinear time?

- **Differential privacy better**: Faster, more accurate.

- **Fair, aware, diverse**: More constraints to favor some specific solutions.

Future Challenges

- **Lower Bound**: What is the best approximation ratio we can get in sublinear time?

- **Differential privacy better**: Faster, more accurate.

- **Fair, aware, diverse**: More constraints to favor some specific solutions.