Coresets for Streaming & Clustering

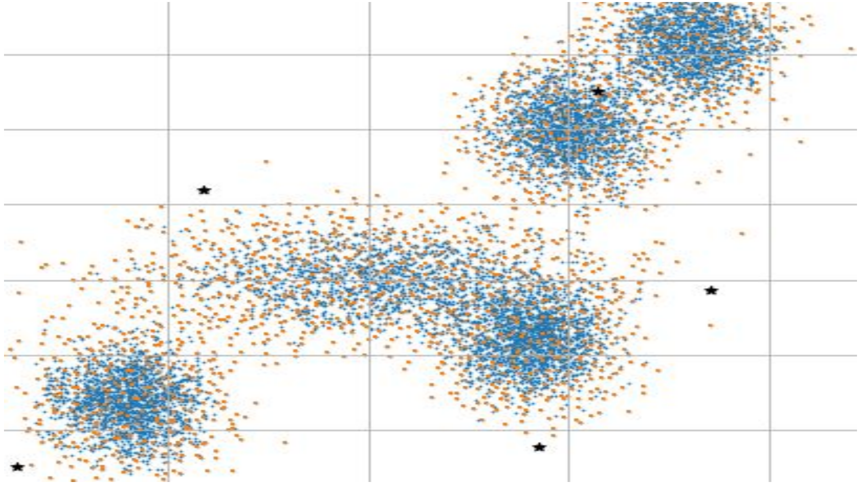
December 09, 2021

Andrew Roberts, Bhushan Suwal, Karan Vombatkere

What is a Coreset?

Given a set of input points P, a Coreset S is a subset of P, such that we can get a good approximation to the original input by solving the problem directly on S.

Coresets are much smaller than the input (typically poly-logarithmic)



Coresets Algorithms Implementation

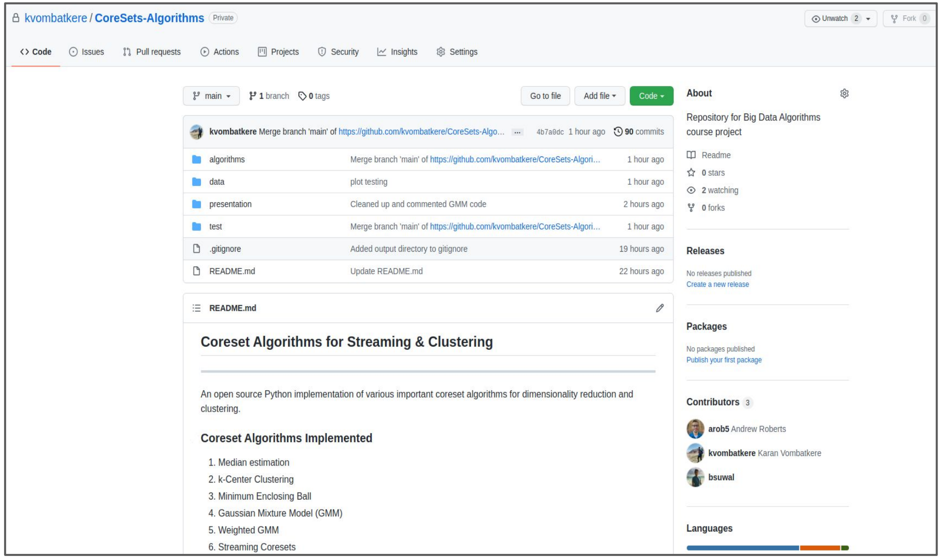
GitHub Repository: https://github.com/kvombatkere/CoreSets-Algorithms

Coreset Algorithms:

- Median Estimation
- Minimum Enclosing Ball (MEB)
- k-center Clustering
- Streaming k-means/k-median
- Gaussian Mixture Models (GMM)
- Weighted GMM

Experiments/Analysis:

- Synthetic, random data
- Real-world datasets

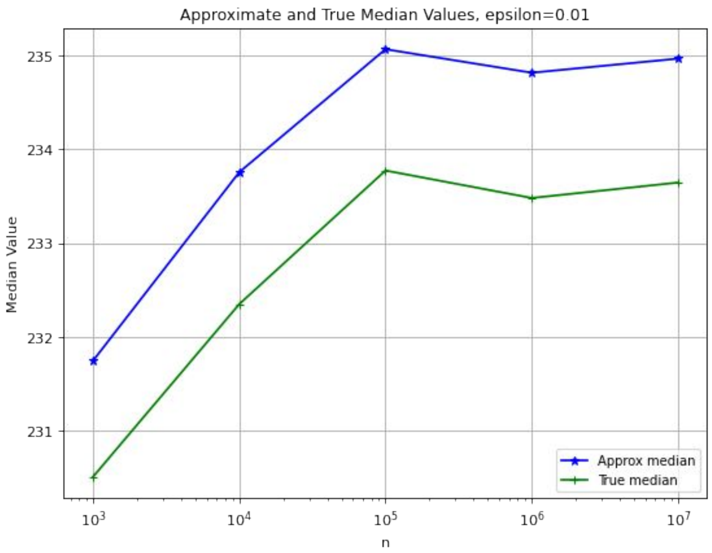


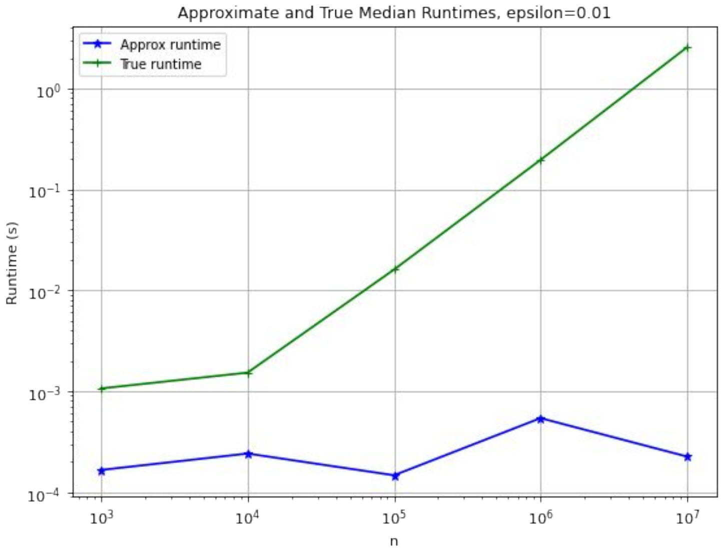
Coreset for Median Estimation

Given sequence of numbers $x_1, \ldots, x_n$ - partition into $O(1/\varepsilon)$ subsequences, compute $\pm\varepsilon n$ approximate median

Analyzed on synthetic data with samples drawn randomly from $\Gamma(k, \theta) = \Gamma(5, 50)$

Runtime comparison with numpy.median()





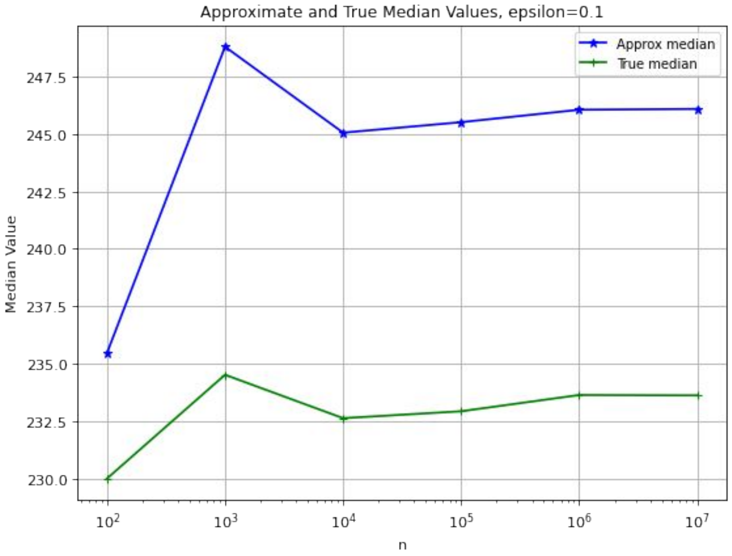
Median Estimation, ε=0.01

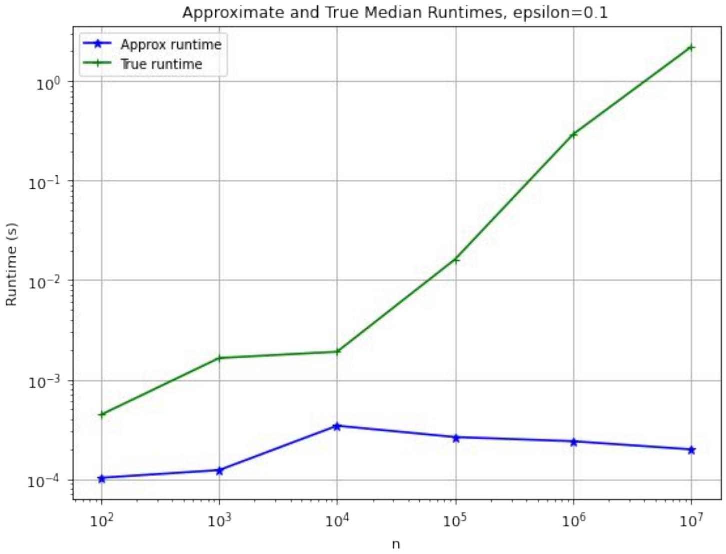
Coreset for Median Estimation

Given sequence of numbers $x_1, \dots, x_n$ - partition into $O(1/\varepsilon)$ subsequences, compute $\pm\varepsilon n$ approximate median

Analyzed on synthetic data with samples drawn randomly from $\Gamma(k, \theta) = \Gamma(5, 50)$

Runtime comparison with numpy.median()





Median Estimation, ε=0.1

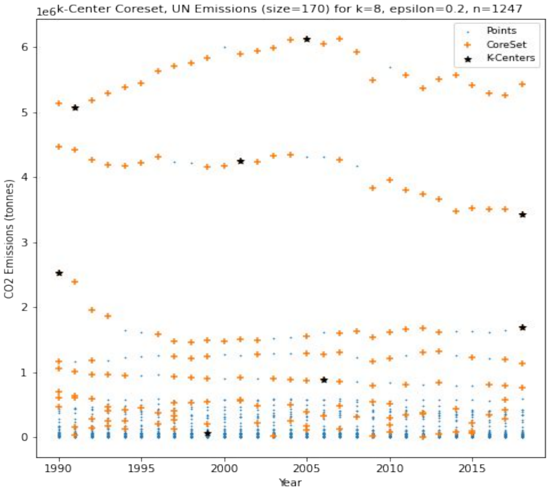
Coresets for k-Clustering

Given a set P of *n* of points in $\mathbb{R}^d$, and an integer $k > 0$, the goal is to partition P into *k* subsets such that a cost function **μ** that measures the extent of a cluster is minimized.

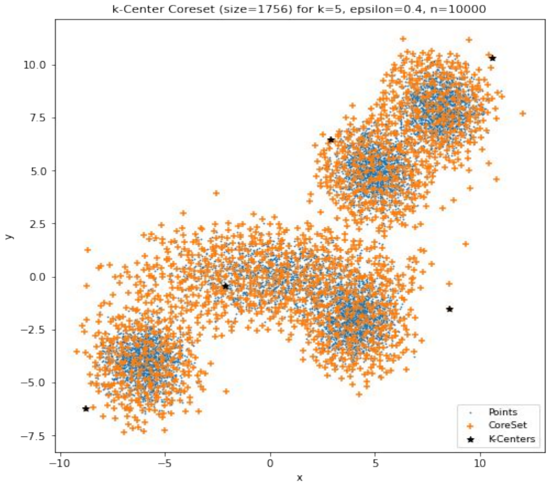
- **k-center**: objective is ***max*** $\boldsymbol{\mu(P_i)}$
- **k-means/k-median**: objective is $\sum \boldsymbol{\mu(P_i)}$

For $0 < \varepsilon < ½$, we compute an additive ε-coreset of size $O(k/\varepsilon^d)$ for k-center clustering.

UN CO_2 Emissions



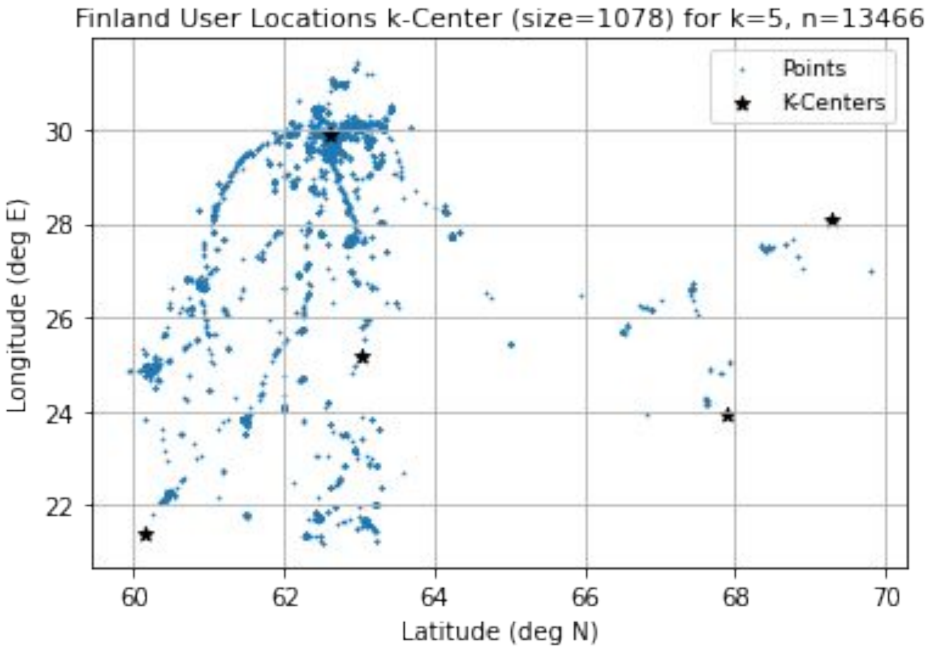
Multivariate Gaussian



Agarwal, Pankaj K., Sariel Har-Peled, and Kasturi R. Varadarajan. "Geometric approximation via coresets." Combinatorial and computational geometry 52.1-30 (2005): 3

k-center Clustering: Experimental Results

Finland 2012 User Locations (MOPSI GPS Data): n=13466

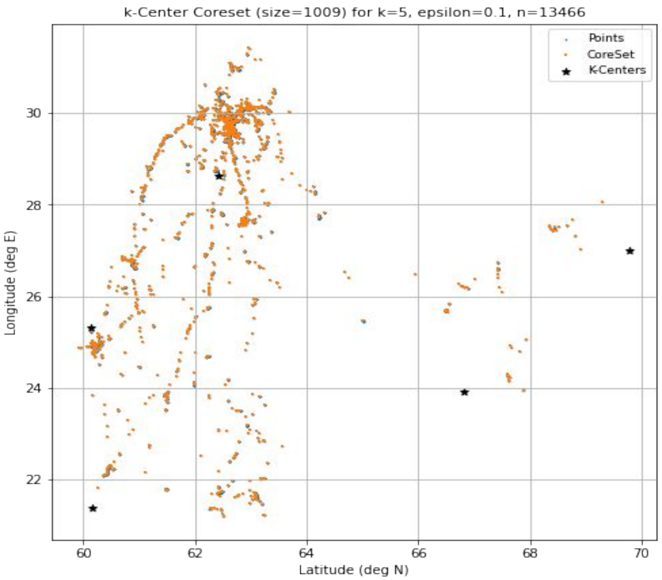


k-center clustering on original data (n=13466)

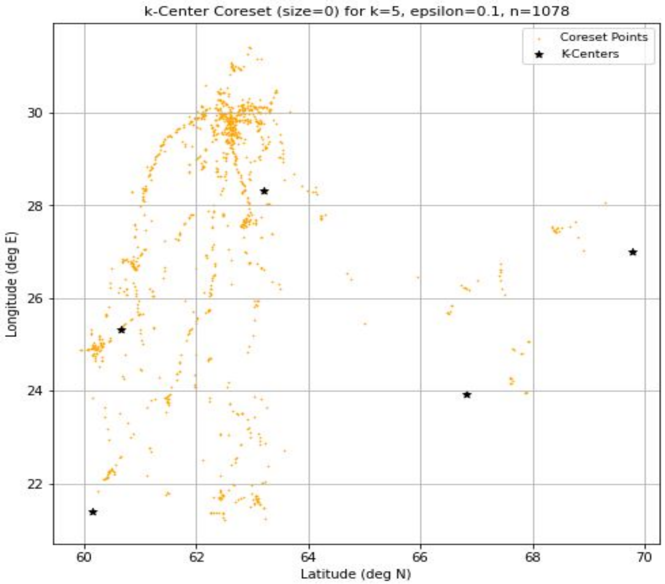
k-center Clustering: Experimental Results

Performed k-center clustering on all *n* points, and then on coreset (50 iterations)

- k-center *(k=5)* average clustering cost = *3.29*
- Coreset k-center *(k=5, ε=0.1)* average clustering cost = *3.36*



k-center coreset and original k-centers



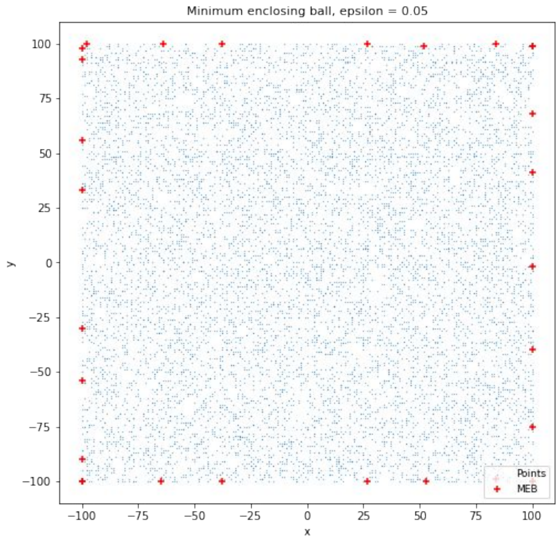
k-center clustering on coreset (n=1078)

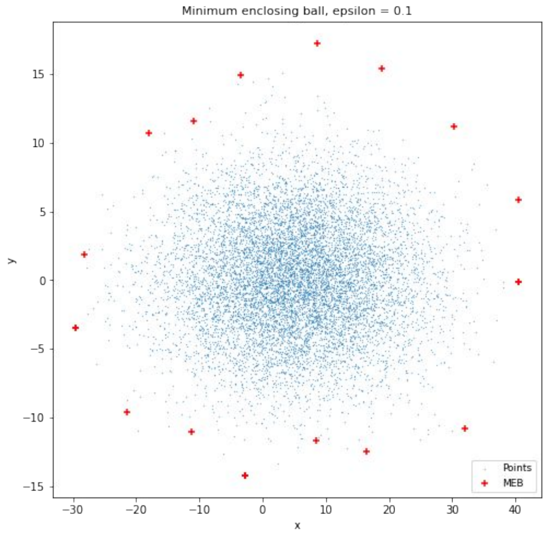
Coreset for Minimum Enclosing Ball (MEB)

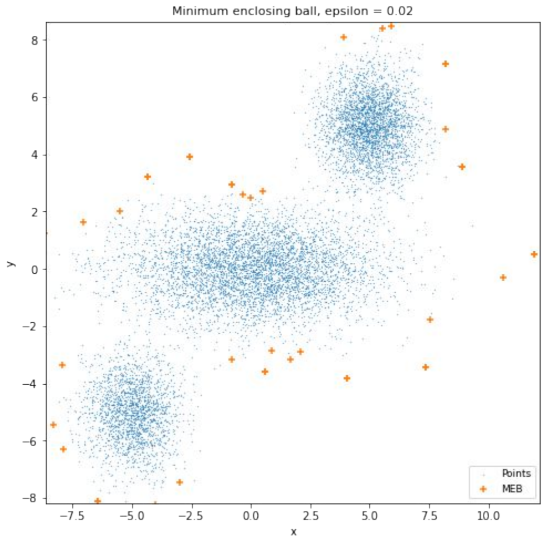
Given a set of points P, the MEB problem consists of finding the smallest ball that encloses the points in P.

Compute a θ-grid consisting of $O(1/\theta^{(d-1)})$ vectors, (1+ε)-coreset of size $O(1/\varepsilon^{(d-1)/2})$

Tested on synthetic data with samples drawn randomly from uniform and gaussian distributions.



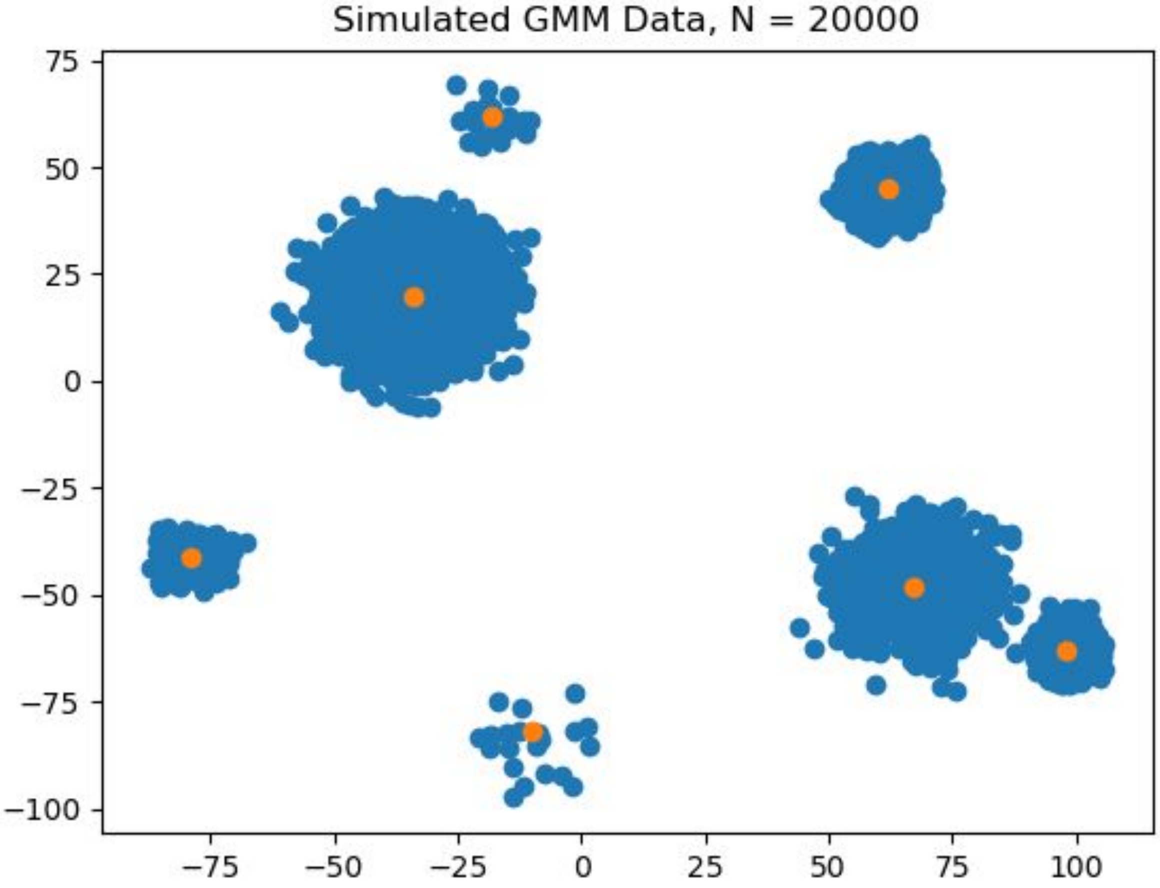




Gaussian Mixture Models (GMMs)

$$p(x_i|\theta) = \sum_{j=1}^{k} w_j \mathcal{N}(x_i|\mu_j, \Sigma_j)$$

$$\theta = (w_1, \mu_1, \Sigma_1, \ldots, w_k, \mu_k, \Sigma_k)$$


Simulated GMM Data, N = 20000
75
50
25
0
−25
−50
−75
−100
−75
−50
−25
0
25
50
75
100

Coresets for GMMs

- K-means approximation

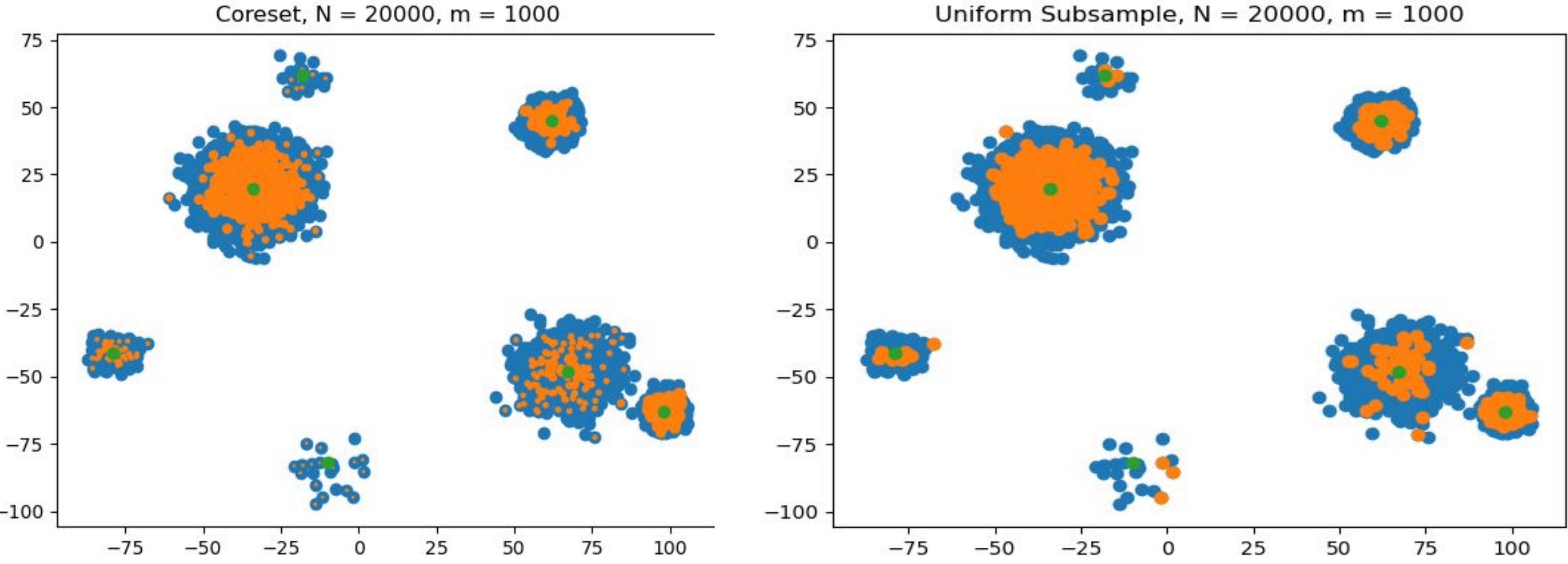
- Importance Sampling

$$\mathcal{L}(C) = \sum_{x \in C} \gamma_x \log p(x|\theta) \approx \sum_{x \in \mathcal{X}} \log p(x|\theta) = \mathcal{L}(\mathcal{X})$$

- Weighted EM Algorithm

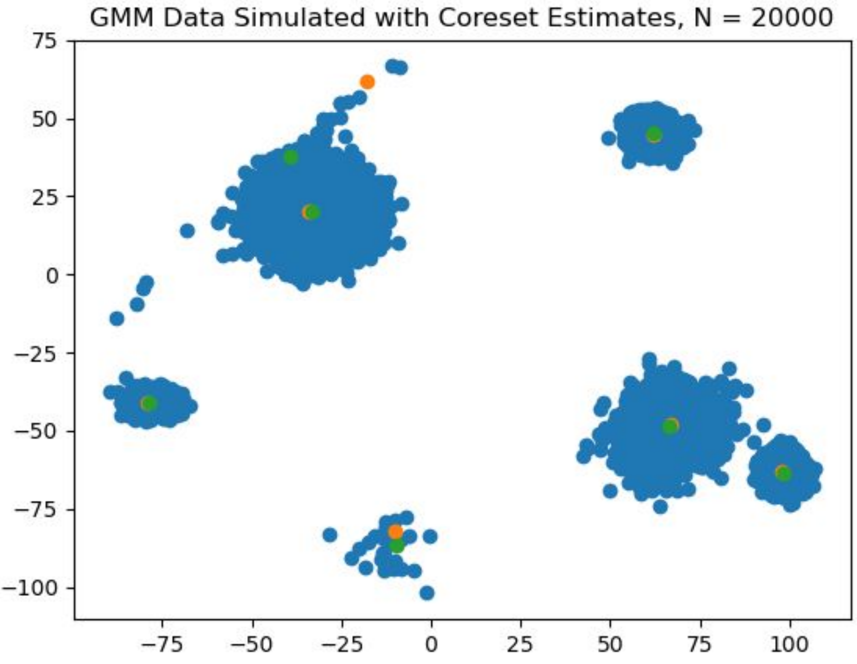
Mario Lucic, Matthew Faulkner, Andreas Krause, and Dan Feldman. “Training Gaussian Mixture Models at Scale via Coresets.” Journal of Machine Learning Research. 2018.

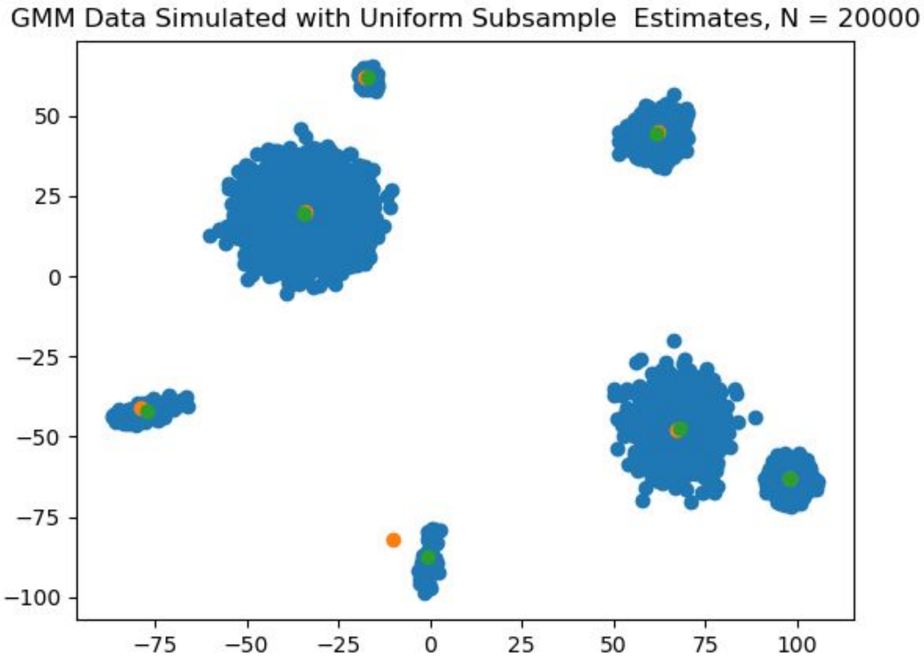
Coresets for GMMs



Mario Lucic, Matthew Faulkner, Andreas Krause, and Dan Feldman. “Training Gaussian Mixture Models at Scale via Coresets.” Journal of Machine Learning Research. 2018.

Coresets for GMMs: Experimental Results





Coresets for Streaming

k-means: objective is to minimize $\sum \mu(P_i)^2$

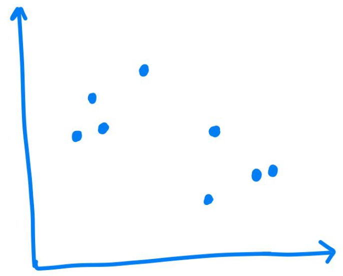
k-medians: objective is to minimize $\sum \mu(P_i)$

The challenge is to maintain a (k, ε) coreset having seen incomplete data.

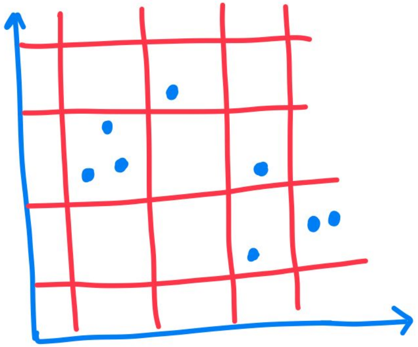
This is motivated by the fact we can’t hold entire dataset in memory.

1. Partition point sequence into chunks P_1 , P_2 , … , P_n
2. Build a d-dimensional grid in the point space.
3. Each box in the grid sends one representative chosen uniformly at random to the coreset Q_i, and the weight of representative is the sum of weights of points in the box.
4. If too many points in coreset, double the side-length of the grid boxes.
5. Coreset $Q = \bigcup Q_i$

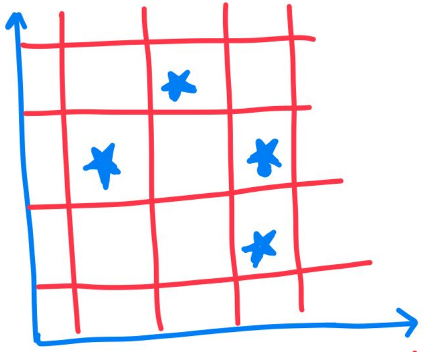
Har-Peled, Sariel, and Soham Mazumdar. "On coresets for k-means and k-median clustering." Proceedings of the thirty-sixth annual ACM symposium on Theory of computing. 2004.



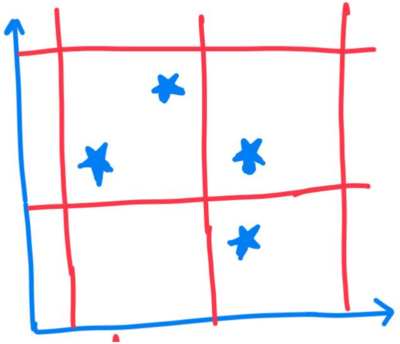
1. Take in points in a chunk P_1



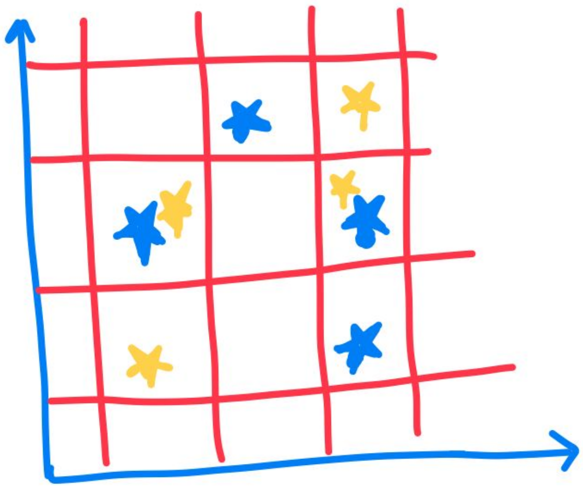
2. Build a grid in the point space



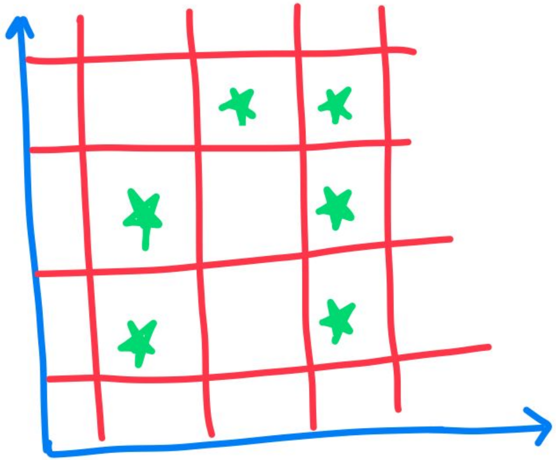
3. Select a representative to be in coreset



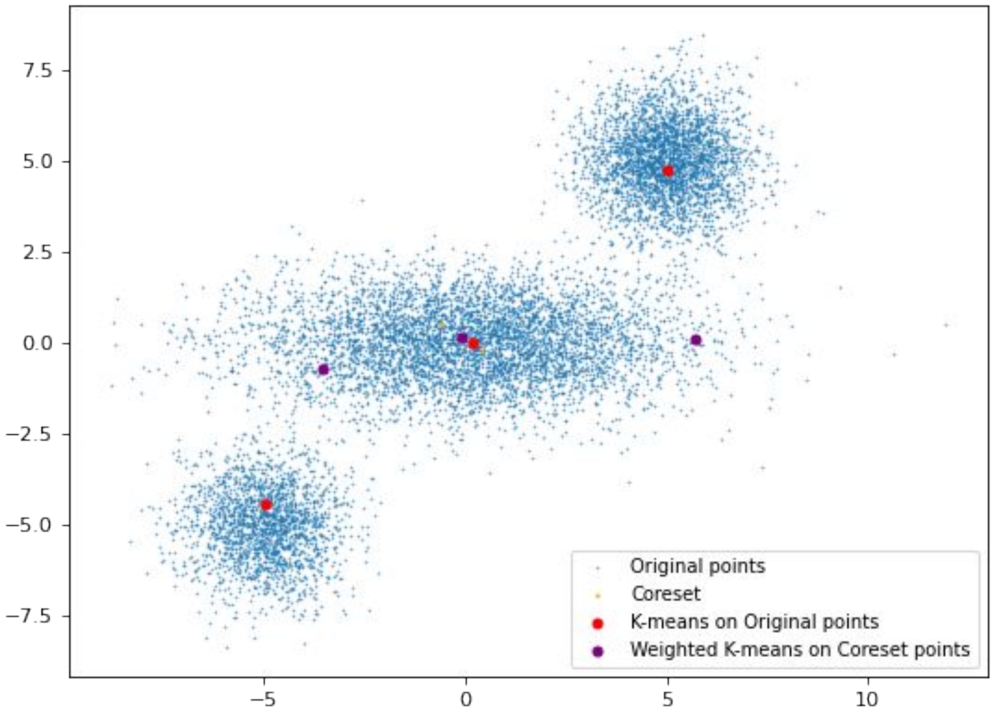
4. Increase grid size if too many points in coreset



5. Take a union of $P_1 \cup P_2$

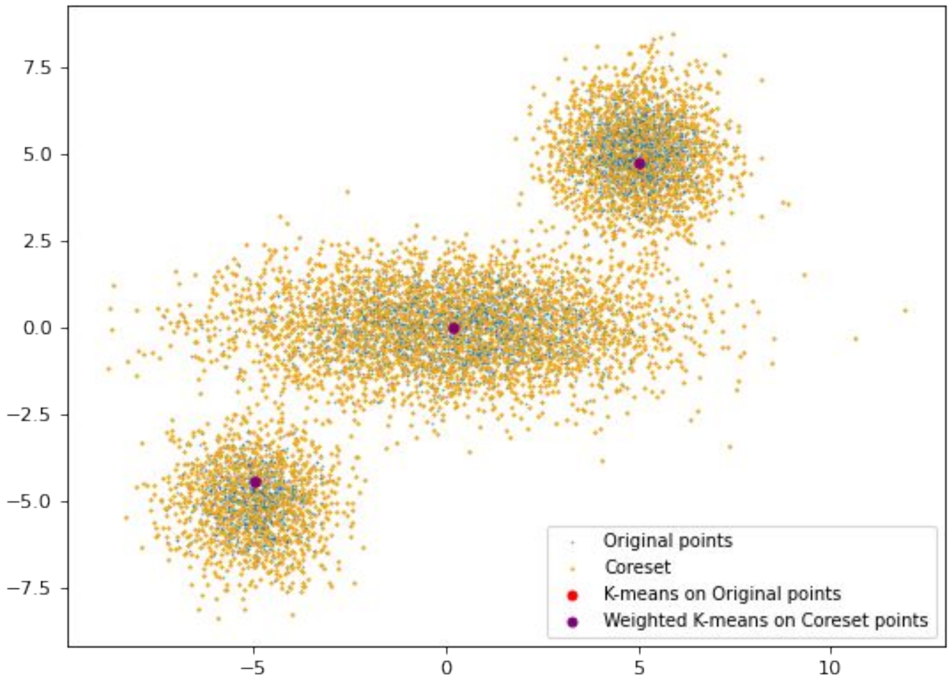


6. Build new coreset



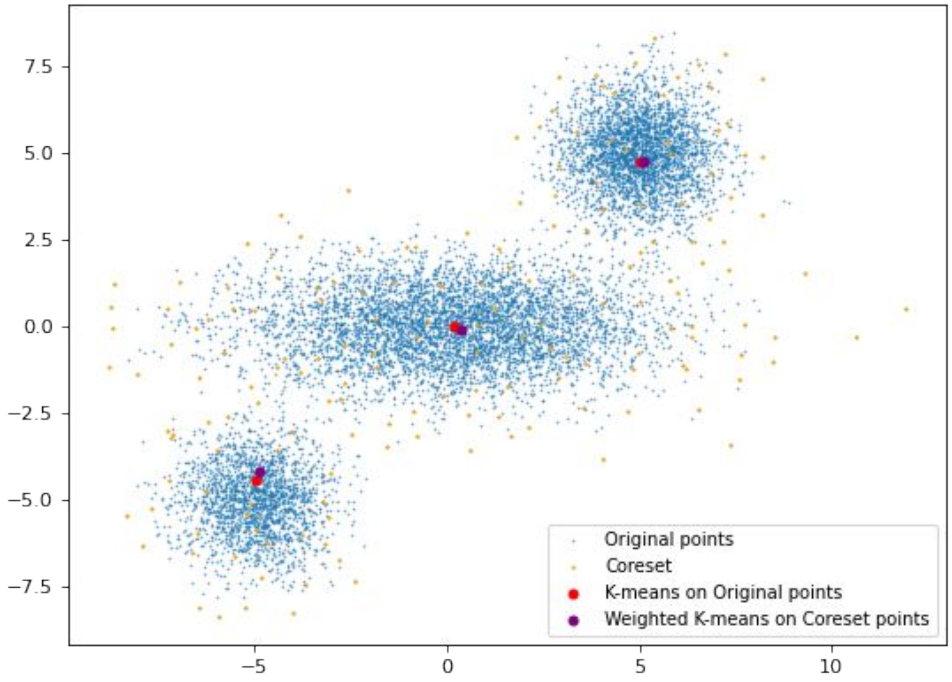
Max Coreset Size: 5
Total Stream Length: 10k
Chunk size: 1k

Grid boxes are too big, results aren’t great.



Max Coreset Size: 7k
Total Stream Length: 10k
Chunk size: 1k

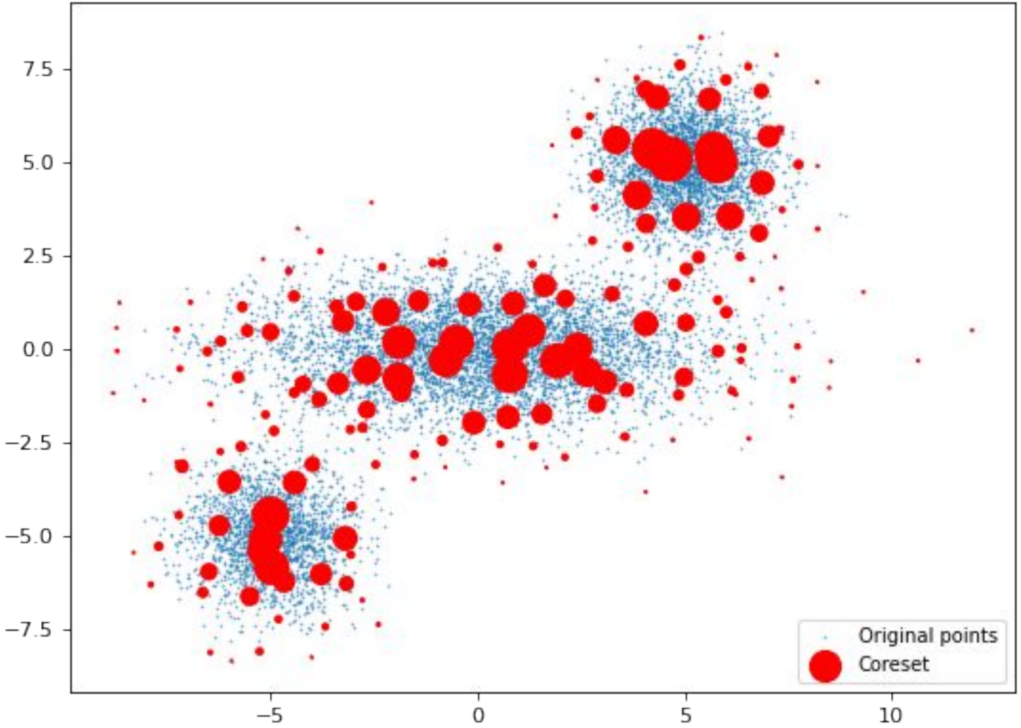
Size of coreset is large,
Results improve.



Max Coreset Size: 500
Total Stream Length: 10k
Chunk size: 1k

Good results with
poly(log) memory usage.

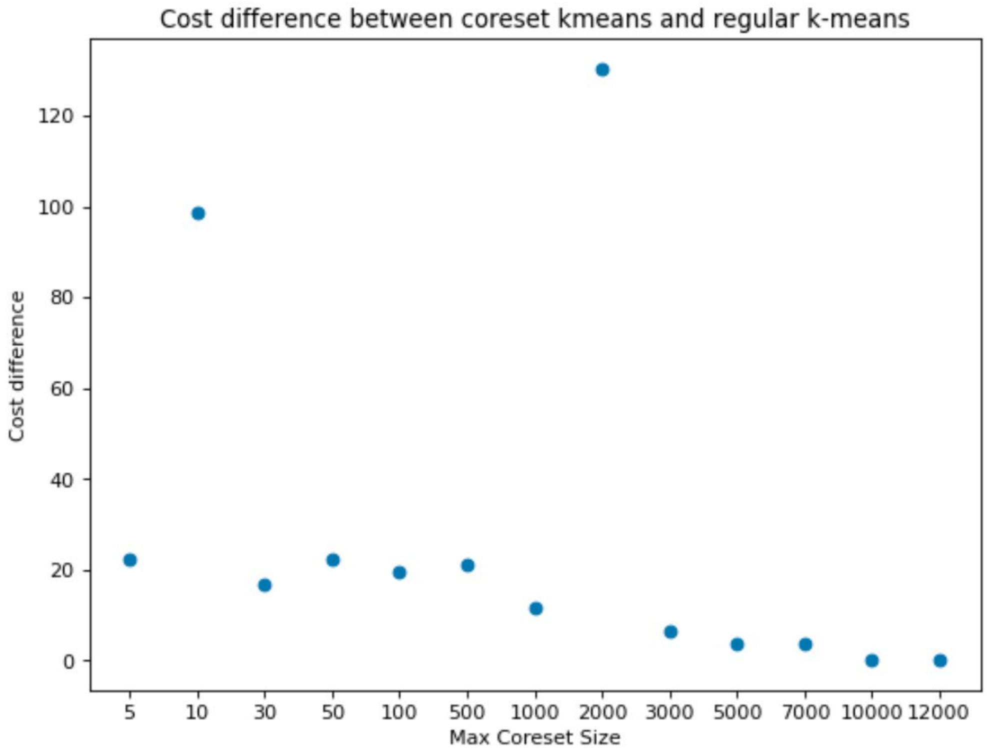
Same coreset, sized by weight.



Max Coreset Size: 500
Total Stream Length: 10k
Chunk size: 1k

Good results with
poly(log) memory usage.

Cost decreases as Coreset size increases (unsurprisingly)



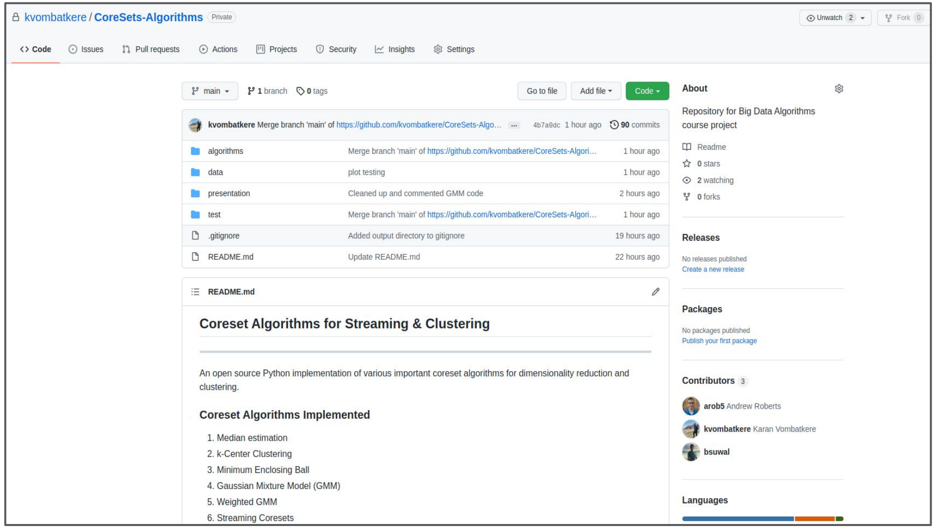
- Each point is the median over 30 runs
- Outlier value at Coreset size 2000 (2x chunk size) points towards relationship between chunk size and cost (??)
- Similar qualitative results for K-medians

Summary

GitHub Repository: https://github.com/kvombatkere/CoreSets-Algorithms

Coreset Algorithms:

- Median Estimation
- Minimum Enclosing Ball (MEB)
- k-center Clustering
- Streaming k-means/k-median
- Gaussian Mixture Models (GMM)
- Weighted GMM



References

1. Agarwal, Pankaj K., Har-Peled, Sariel, and Kasturi R. Varadarajan. "Geometric approximation via coresets." Combinatorial and computational geometry 52.1-30 (2005): 3.
2. Feldman, Dan, Matthew Faulkner, and Andreas Krause. "Scalable Training of Mixture Models via Coresets." NIPS. 2011.
3. Har-Peled, Sariel, and Soham Mazumdar. "On coresets for k-means and k-median clustering." Proceedings of the thirty-sixth annual ACM symposium on Theory of computing. 2004.
4. Mario Lucic, Matthew Faulkner, Andreas Krause, and Dan Feldman. “Training Gaussian Mixture Models at Scale via Coresets.” Journal of Machine Learning Research. 2018.