Coresets for k-Means and k-Median Clustering and their Applications

Sariel Har-Peled and Soham Mazumdar

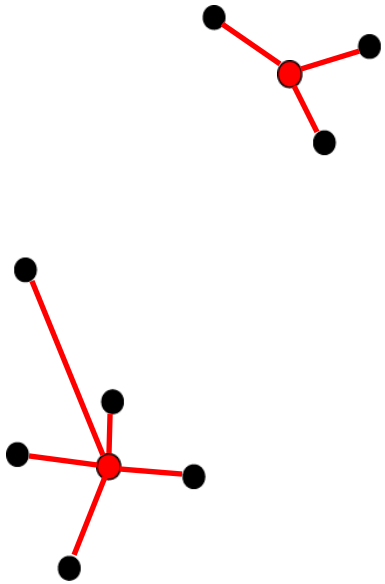
March 8, 2006

Problem Introduction

- We are given a point set *P* in R^d of size *n*
- Find a set of *k* points C such that the cost function is minimized
- Cost functions
  - Median: $\sum_{p \in P} w(p) d(p, C)$ $\quad d(p, X) = \min_{x \in X} ||xp||$
  - Discrete median: $\sum_{p \in P} w(p) d(p, C)$ $\quad C \in P$
  - Mean: $\sum_{p \in P} w(p) (d(p, C))^2$
- Streaming

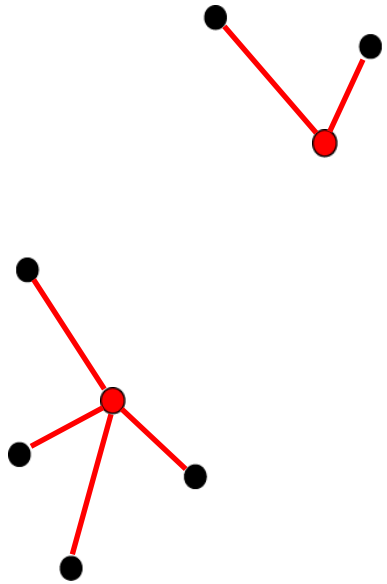
Costs

k-medians

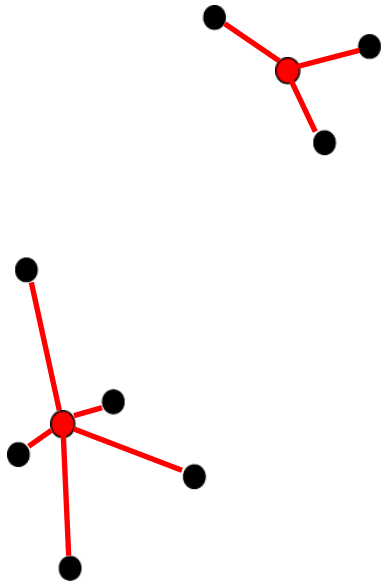


$$\sum_{p \in P} w(p) d(p, C)$$

Discrete k-medians



k-means



$$\sum_{p \in P} w(p) (d(p, C))^2$$

Results

- Builds on the algorithms we saw last week
  - Kolliopoulos and Rao [KR99]
  - Matoušek [Mat00]
- Results
  - k-median

$O(\rho n(\log n)\log k)$ $\rho = exp[O((1+\log 1/\epsilon)/\epsilon)^{d-1}]$
$O(n + \rho k^{O(1)} \log^{O(1)} n)$

  - Discrete k-median

$O(\rho n \log n \log k)$
$O(n + \rho k^{O(1)} \log^{O(1)} n)$

  - k-mean

$O(n(\log n)^k \epsilon^{-2k^2 d})$
$O(n + k^{k+2}\epsilon^{-(2d+1)k} \log^{k+1} n \log^k \frac{1}{\epsilon})$

Overview

- Similar for k-medians and k-means
- Construct a series of sets

$$P \Rightarrow A \Rightarrow S \Rightarrow D \Rightarrow C$$

- Algorithm Components
  - P: Point set
  - S: Coreset
  - A: Constant factor approximation
  - D: Centroid set
  - C: k centers

Coresets for k-median

- Definition: S is an (k,ε)-coreset if

$$(1-\epsilon)\nu_C(P) \leq \nu_C(S) \leq (1+\epsilon)\nu_C(P)$$

- Construction
- Begin with P and A where

$$\nu_A(P) \leq c\nu_{opt}(P,k)$$

- Estimate average radius

$$R = \nu_A(P)/(cn)$$

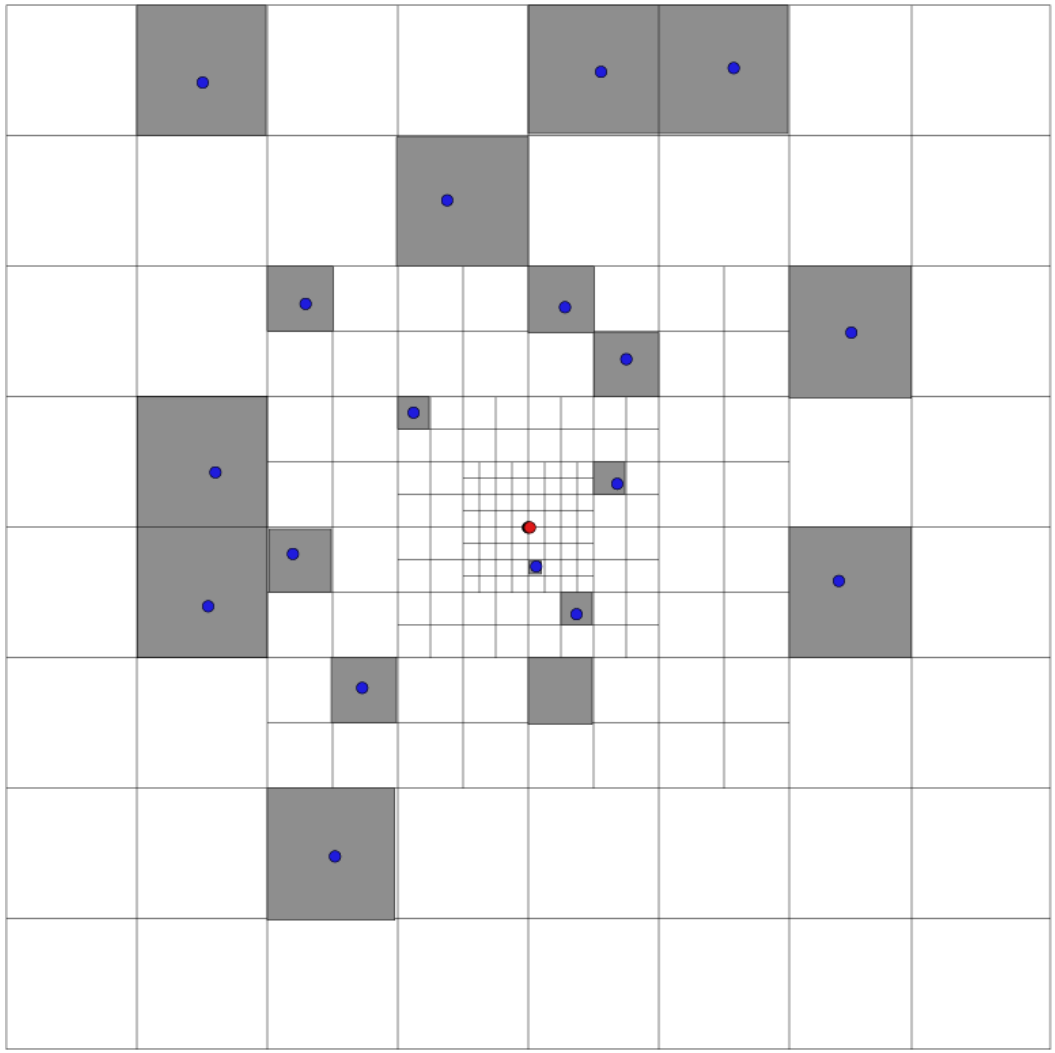
$$||px_i|| \leq \nu_A(P) \leq cnR$$

- Exponential grid around x 2 A with M levels

$$M = \lceil 2\log(cn) \rceil$$

Exponential Grid

- For each point in A
- Level j has side length $\varepsilon R2^j/(10cd)$
- Pick a point in each non-empty cell
- Assign weight by number of points in cell



Cost of Constructing S

- Size

$$|S| = O((|A| \log n)/\epsilon^d)$$

- In each level, constant number of cells
  - log n levels
- Cost of construction
  - Constant factor approximation to cost $\nu_A(P)$
  - Nearest Neighbor queries

NN Queries

Naïve: O(mn)

[AMN+98]: O(log m) after O(m log m)

Here: $O(n+mn^{1/4} \log n)$

Total Cost

If m = $O(\sqrt{n})$

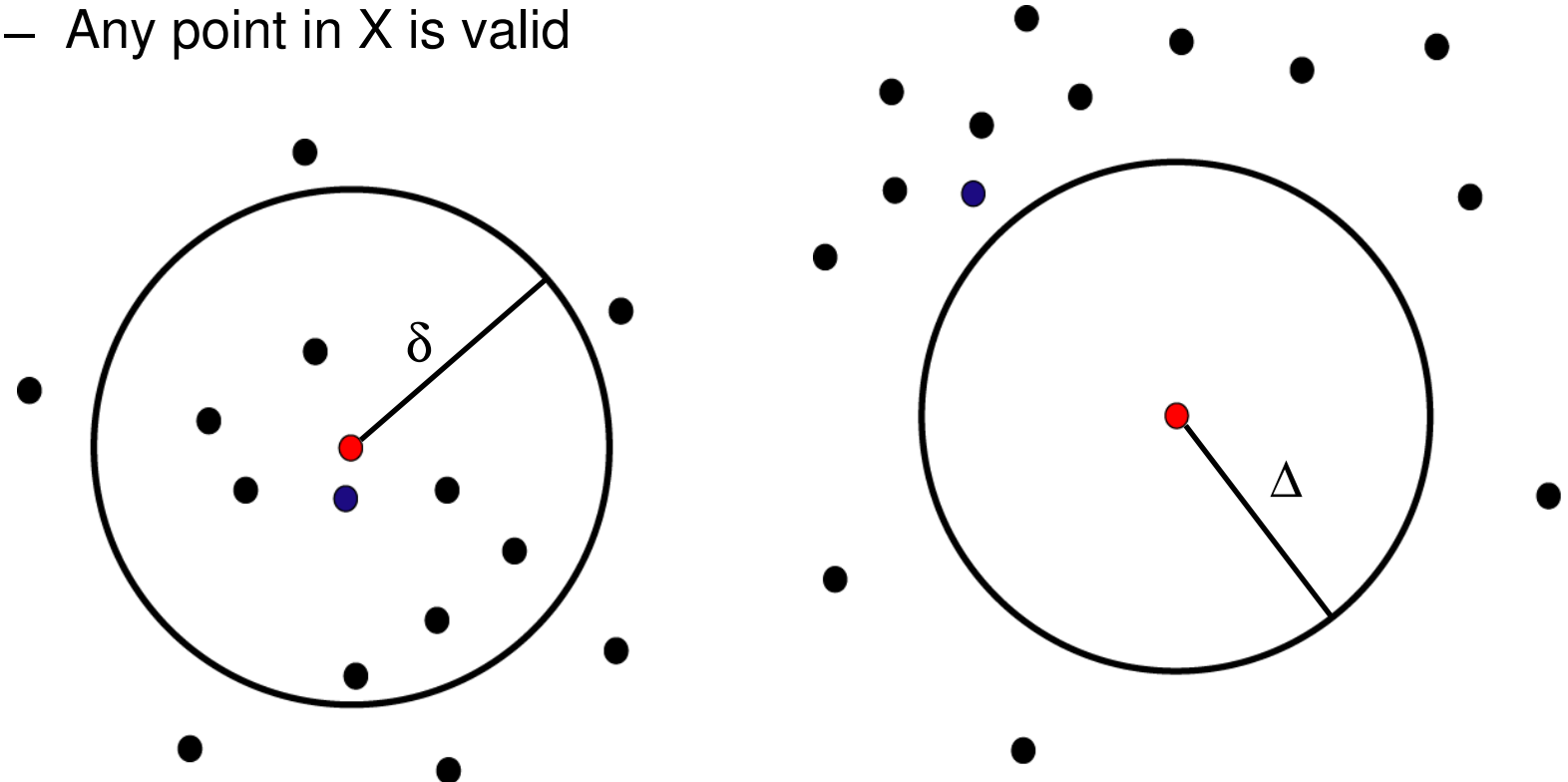
$$O(n + mn^{1/4} \log n) = O(n)$$

else

$$O(m \log m + n \log m + n) = O(n \log m)$$

Fuzzy Nearest Neighbor Search in O(1)

- ε-approximate nearest neighbors to a set X
- If distance $q < \delta$
  - Any point in X which is closer than δ is valid
- If $q > \Delta$
  - Any point in X is valid



Proof of Correctness

- p 2 P and its image in S ! p'
- For any k points (Y) the error is

$$
\begin{aligned}
E &= |\nu_Y(P) - \nu_Y(S)| \leq \sum_{p \in P} |d(p, Y) - d(p', Y)| \\
&\leq \sum_{p \in P} ||pp'|| = \sum_{d(p,A) \leq R} (||pp'|| + \sum_{d(p,A) > R} (||pp'|| \\
&\leq \frac{\epsilon}{10c} nR + \frac{\epsilon}{10c} \sum_{p \in P} (d(p, A)) \leq \frac{2\epsilon}{10c} \nu_A(P) \\
&\leq \epsilon \nu_{opt}(P, k)
\end{aligned}
$$

Coresets for k-means

- Similar to k-medians
- Lower bound estimate for average mean radius

$$R = \sqrt{\mu_A(P)/cn} \qquad R_{opt}(P,k) = \sqrt{\mu_{opt}(P,k)/n}$$

- A is a constant factor approximation

$$\mu_A(P) \leq c\mu_{opt}(P,k)$$

- Using R and A, we construct S with the exponential grid
- Size: $|S| = O((m \log n)/(c\epsilon)^d)$

- Running time: $O(n)$ if $m = O(n^{1/4})$
  else $O(n \log n)$

Proof of Correctness

- Idea: Partition P into 3 sets
  - Points that are close to A and B – small error
  - Points closer to B than to A – ε fraction error
  - Points closer to A than to B – "better" than optimal
- Bound each error

$$E = |\mu_B(P) - \mu_B(S)| \leq \sum_{p \in P} |d(p,Y)^2 - d(p',Y)^2|$$
$$\leq \sum_{p \in P} |(d(p,Y) - d(p',Y))(d(p,Y) + d(p',Y))|$$

- Result: $E = |\mu_B(P) - \mu_B(S)| \leq$
$$\leq E_R + E_A + E_B \leq \frac{3\epsilon}{3}\mu_B(P)$$

Errors

$$\begin{aligned}
\mathcal{E}_R &= \sum_{p\in P_R} \left|\big(\mathbf{d}(p,B)-\mathbf{d}(p',B)\big)\big(\mathbf{d}(p,B)+\mathbf{d}(p',B)\big)\right| \\
&\le \sum_{p\in P_R} \|pp'\|(2\mathbf{d}(p,B)+\|pp'\|) \le \sum_{p\in P_R} \frac{\varepsilon}{10}R\Big(2R+\frac{\varepsilon}{10}R\Big) \\
&\le \frac{\varepsilon}{3}\sum_{p\in P_R} R^2 \le \frac{\varepsilon}{3}\mu_{\mathrm{opt}}(P,k),
\end{aligned}$$

$$\begin{aligned}
\mathcal{E}_A &= \sum_{p\in P_A} \|pp'\|(2\mathbf{d}(p,B)+\|pp'\|) \le \sum_{p\in P_A} \frac{\varepsilon}{10c}\mathbf{d}(p,A)\Big(2+\frac{\varepsilon}{10c}\Big)\,\mathbf{d}(p,A) \\
&\le \frac{\varepsilon}{3c}\sum_{p\in P_A} (\mathbf{d}(p,A))^2 \le \frac{\varepsilon}{3}\mu_{\mathrm{opt}}(P,k) \le \frac{\varepsilon}{3}\mu_B(P).
\end{aligned}$$

$$\begin{aligned}
\mathcal{E}_B &= \sum_{p\in P_B} \|pp'\|(2\mathbf{d}(p,B)+\|pp'\|) \le \sum_{p\in P_B} \frac{\varepsilon}{10c}\mathbf{d}(p,B)\Big(2\mathbf{d}(p,B)+\frac{\varepsilon}{10c}\mathbf{d}(p,B)\Big) \\
&\le \sum_{p\in P_B} \frac{\varepsilon}{3}\mathbf{d}(p,B)^2 \le \frac{\varepsilon}{3}\mu_B(P).
\end{aligned}$$

Fast Constant Factor Approximation

- In both cases need constant approximation
  - i.e. set A
- Use more than *k* centers – $O(k \log^3 n)$
- Good for both k-means and k-medians
- 2-approximate clustering (min-max clustering)
  - $k = O(n^{1/4})$! $O(n)$ [Har01a]
  - $k = \Omega(n^{1/4})$! $O(n \log k)$ [FG88]

Picking Sets

- Distance between points in V at least L
- L is an estimate of cost

$$(L/2)^2 \leq \mu_{opt}(V,k) \leq \mu_{opt}(P,k) \leq nL^2$$

- Y is a random sample of P
  - size $\rho = \gamma\, k \log^2 n$
- Desired set of centers ! X = Y U V
- We want a large “good” subset for X
- “Good” defined in terms of bad points

Bad Points

- Definition
  A point is bad with respect to a set X if the cost is much larger than the optimal center would pay. More precisely

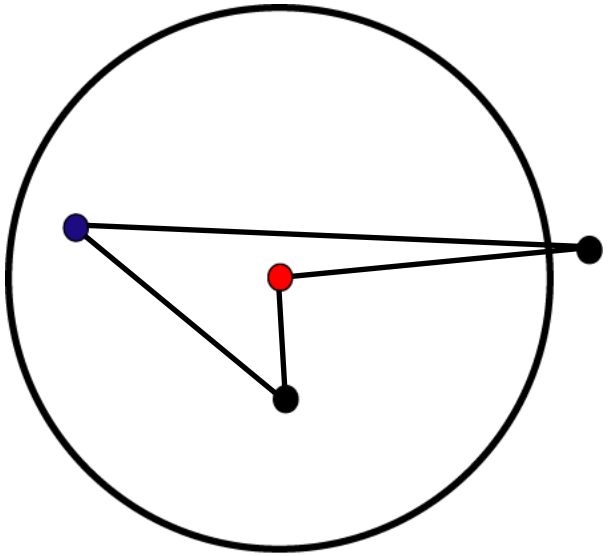
$$d(p, X) \geq 2d(p, C_{opt})$$

- There are few bad points in X
- There contribution to the clustering cost is small

Few Bad Points

- C_{opt} are optimal center k-means
- Place ball b_i around each point c_i
- Each ball contains $\eta = n/(20k \log n)$ points
- Choose γ so at least one x_i in b_i
- Any p outside b_i is not a bad point

$$||px_i|| \leq 2||pc_i||$$



- Number of bad points

$$\beta = k\eta = n/(20logn)$$

Clustering Cost of Bad Points

- Hard to determine set of bad point
- For every point in P, compute approximate nearest neighbor in X
  - Cost is same as in construction of S
- Partition P

$$P[a,b] = \left\{ p \in P \;\middle|\; a \le r(p) < b \right\}$$

- Good set P'
  - P_α is the last class more than 2 β points
  - P' = U P_i for i =1…α
  - |P'| ¸ n/2 and $\mu_X(P') = O(\mu_{C_{opt}}(P'))$

Proof

- Size of P': $|P'| \geq n/2$

  $|P'| \leq (n - |P_\infty| - M(2n/20 \log n))$

  $P_\infty \subseteq P_{bad} \qquad |P_{bad}| \leq \beta$

- Cost is roughly the same for all p'

  $\mu_X(P') \leq 20\mu_{C_{opt}}(P')$

- Constant factor k-median clustering
  - Run O(log n) iterations
  - In each iteration we get |X| = O (k $\log^2$ n)
  - So total number of centers O(k $\log^3$ n)
  - Approximation bounded by $\mu_{C_{opt}}(P') \geq \mu_X(P')/32$

(1+ε) k-Median Approximation

- Make A of size $O(k \log^3 n)$
- Get coreset S of size $O(k \log^4 n)$
- Compute O(n) approximation using k-center (min-max) algorithm [Gon85]
  - Result is C_0
- Use local search to get down to exactly k centers [AGK+01]
  - Swap a point in the set of centers with one outside
  - Keep it if it shows considerable improvement
- Use these with exponential grid once more to get the final coreset S
- Time: $O(|S|^2 k^3 \log^9 n)$
- Size: $O((k/\varepsilon^d) \log n)$

Centroid Sets

- To apply [KR99] directly but only works in discrete case
- Create a centroid set
  - Make a (k,ε/12)-coreset S
  - Compute exponential grid around each point in S with R = ν_B (P)/n
  - Centroid set D size of $O(k^2 \varepsilon^{-2d} \log^2 n)$
- Proof

$$\begin{aligned} \left|\nu_X(P) - \nu_{C_{\text{opt}}}(P)\right| &\leq \left|\nu_X(P) - \nu_X(\mathcal{S})\right| + \left|\nu_X(\mathcal{S}) - \nu_{C_{\text{opt}}}(\mathcal{S})\right| \\ &\quad + \left|\nu_{C_{\text{opt}}}(\mathcal{S}) - \nu_{C_{\text{opt}}}(P)\right| \\ &\leq \left(\frac{\varepsilon}{3} + \frac{\varepsilon}{6} + \frac{\varepsilon}{12}\right) \nu_{C_{\text{opt}}}(P) \leq \varepsilon \nu_{C_{\text{opt}}}(P). \end{aligned}$$

- Now run [KR99], using only centers from D

Summary of Construction

- Compute 2-approximate k-center clustering of P
- Compute set of good points P' and X
  - Repeat log n times to get A
- Compute S from A and P using exp. grid
- Compute O(n) approximation of S
- Apply local search alg. to find k centers
- Compute coreset from k centers and P using exp. grid
- Compute D from coreset and k centers using exp. grid
- Apply [KR99] using only centers from D

Discrete k-medians

- Compute $\varepsilon/4$ centroid
- Find representative set
  - Points P snapped to D
  - Discrete centroid set
- Result

$$O\left(n + k^5 \log^9 n + \varrho k^2 \log^5 n\right)$$

k-Means

- Everything is the same up to local search algorithm
- Algorithm due to Kanungo *et al.* [KMN+02]
- Use Maousek [Mat00] to compute k-means on the coreset
- Result

$$O\left(n + k^5 \log^9 n + k^{k+2} \varepsilon^{-(2d+1)k} \log^{k+1} n \log^k (1/\varepsilon)\right)$$

Streaming

- Partition P into sets
  - P_i is empty
  - $|P_i| = 2^i M$ where $M=O(k/\varepsilon^d)$
- Store coreset for each P_j ! Q_j
- Q_j is a (k,δ_j)-coreset for P_i

$$1 + \delta_j \leq 1 + \epsilon/2$$

- U Q_j is a $(k,\varepsilon/2)$-coreset for P
- When new point enters
  - Add new p to P_0
  - If Q_1 exists, merge the two, calculate new coreset and continue until Q_r does not exist
  - Can merge coresets efficiently

End