On approximate geometric *k*-clustering

Jiri Matousek

DCG 2000

Problem

- Given *n*-point set $X \subset R^d$ and *k>1* find a partition $\Pi = (S_1, S_2, \ldots, S_k)$ of minimum cost
- "Geometric" cost function

$$\mathsf{cost}(\Pi) = \sum_{i=1}^{k} \mathsf{cost}(S_i)$$

$$\mathsf{cost}(S) = \sum_{x \in S} ||x - c(S)||^2$$

- Approximately optimal

$$\mathsf{cost}(\Pi) \leq (1+\epsilon)\mathsf{cost}(\Pi')$$

Results

- 2-clustering
  - *n log n* approximate algorithm for fixed $\varepsilon > 0$

$$O\left(n \log n \cdot \varepsilon^{-2d} \log \tfrac{1}{\varepsilon} + n\varepsilon^{-(4d-2)} \log \tfrac{1}{\varepsilon}\right)$$

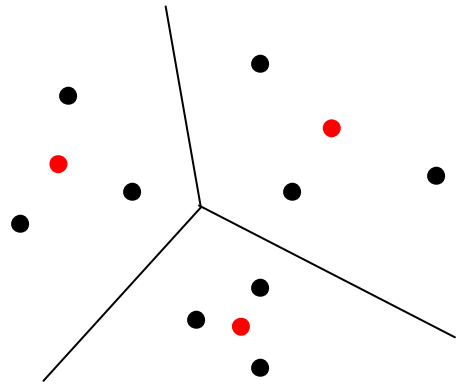
- *k*-clustering

$$O\left(n(\log n)^k \varepsilon^{-2k^2 d}\right)$$

- Can be improved with known lower bound on cluster size

Really Easy Problem

- Given X and cluster centers c_1, c_2, …, c_k, find the optimal clustering of X



General Approach

- Snap *X* to a grid
- Cover the space with points
  - potential cluster centers
  - polynomial in *n*
- Test subsets of size *k* to find the best centers
  - only test sufficiently different subsets
    - Aiming for a “near linear” algorithm

Polynomial Grid

- Thm: Let d and k_0 be fixed. Suppose there is an algorithm A that, for a given $\varepsilon>0$, $k \leq k_0$, and an n-point multiset $X' \subset R^d$ with points lying on an integer grid of size $O(n^3/\varepsilon)$, finds an $(1+\varepsilon)$-approximately optimal k-clustering of X'. Then a $(1+\varepsilon)$-approximately optimal k_0 clustering for an arbitrary n-point subset $X \subset R^d$ can be computed with O(n log n) preprocessing and with at most C calls to algorithm A, with various at most n-point sets X', with $k \leq k_0$, and with $\alpha\varepsilon$ instead of ε, where $\alpha>0$ and C are constants.
- Pf: Grid size $\delta = \alpha\varepsilon\Delta/5n^2$, $\Delta = \text{diam}(X)/n$. Let X be the original and X' the snapped set. If Π is the clustering of X, and Π' is corresponding clustering of X'
  - max change in distance to cluster center $|\text{diam}(X)^2 - (\text{diam}(X) + 2\,\delta)^2| \leq 5\,\delta\,\text{diam}(X)$
  - $|cost(\Pi) - cost(\Pi')| \leq n \cdot 5\delta\ diam(X) \leq \alpha\epsilon\Delta^2$
  - If $\text{cost}(\Pi') > 1/20\Delta^2$, since Π' is $(1+\alpha\varepsilon)$-approximate for X', then corresponding Π is $(1+\varepsilon)$-approximate for X
  - Otherwise, apply the algorithm recursively groups of nearby clusters

Approximate Centroid Sets

- Dfn: C is an ε-approximate centroid set for X if it intersects ε-tolerance ball of every subset of X (of size at least s)
  - ε-tolerance ball for S is centered at c(S), with radius $\frac{\epsilon}{3}\rho(S)$

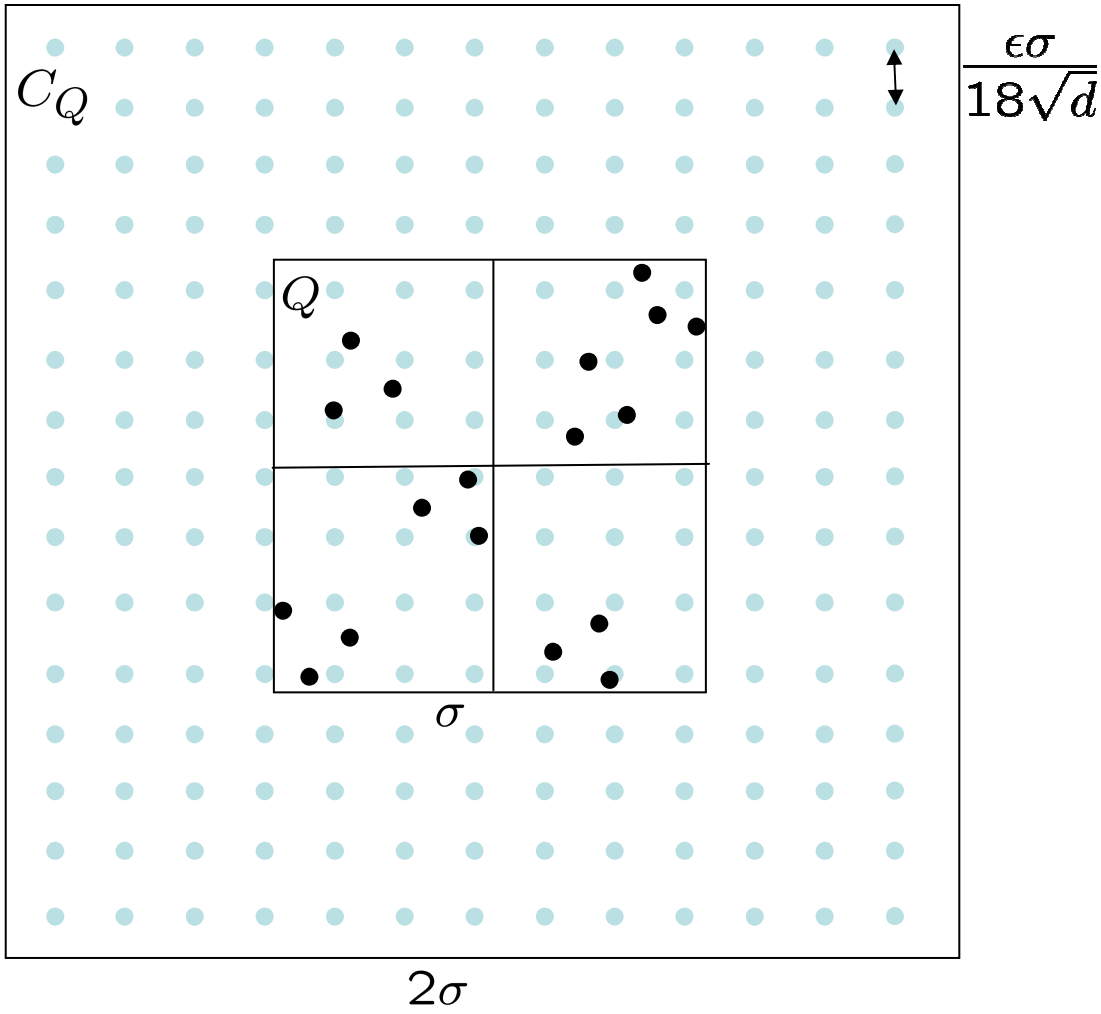
$$\rho(S) = \left(\frac{1}{|S|}\sum_{x \in S} ||x - c(S)||^2\right)^{\frac{1}{2}}$$

- Thm: Let $X \subset R^d$ be a finite point set and let $k \geq 2$, and let C be an ε–approximate centroid set for S with cluster size $\geq s$. Then there are $c_1, c_2, \ldots, c_k \in C$ s.t.

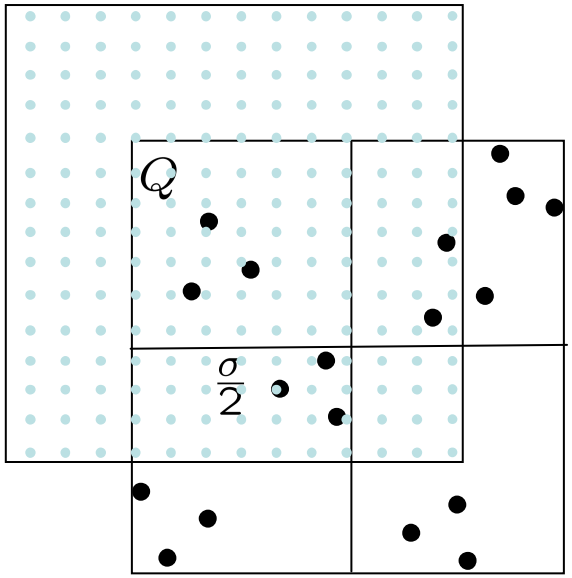
$$\mathsf{cost}(\Pi_{Vor}(c_1, c_2, \ldots, c_k)) \leq (1 + \epsilon)\mathsf{cost}(\Pi)$$

  for all k-clusterings of with all clusters of size at least s
- Pf: By algebra using definition of centroid sets

Contruction



Contruction

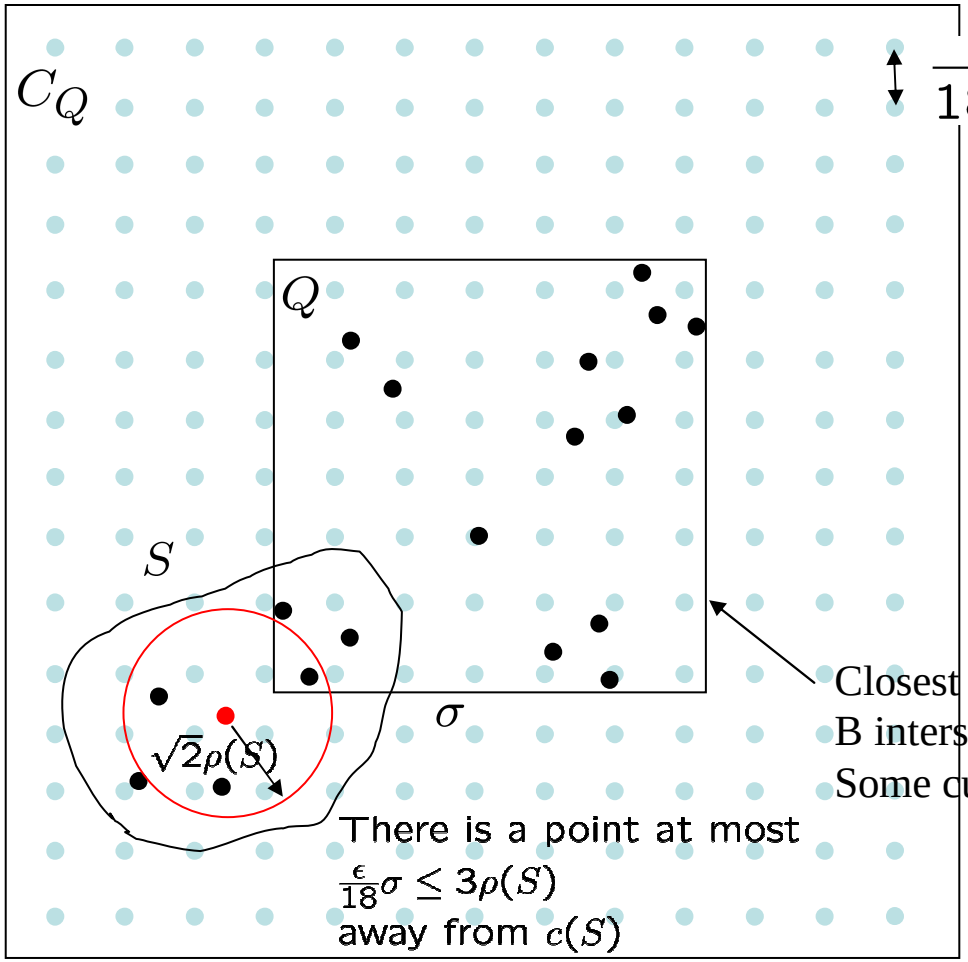


Subdivide as long as

$$\sigma \geq \frac{1}{\sqrt{2}}\delta$$

Q contains at least s/2d+1 points

Contruction



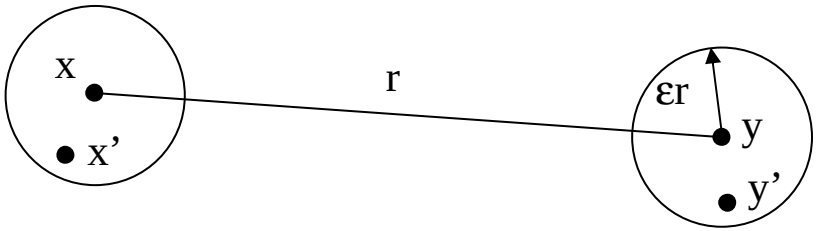
Closest σ larger than diam(B)
B intersects at most 2^d cubes
Some cube has $s/2^{d+1}$ points

Centroid Sets

- Thm: $|C| = O(\frac{n}{s}\epsilon^{-d} \log \frac{nR}{\delta})$ and the construction can be performed in time $O((n + \frac{n}{s}\epsilon^{-d}) \log \frac{nR}{\delta})$
- Pf: There are at most $O(\log \frac{R}{r})$ different side lengths, and each cube contains at least $\frac{s}{2^{d+1}}$ points, so there are $O(\frac{n}{s} \log \frac{R}{r})$

Well-separated Pairs

- Dfn: (x, y) and (x', y') are ε -near if $||x - x'|| \leq \epsilon \cdot ||x - y||$ and $||y - y'|| \leq \epsilon \cdot ||x - y||$

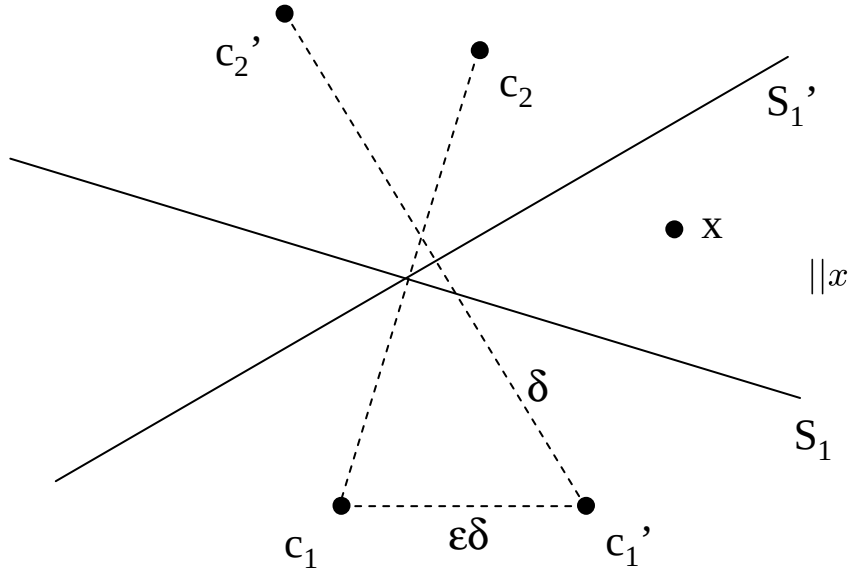


- Set P is ε –separated if no two pairs are ε –near
- P is ε –complete for X if for every pair in X, there is an ε –near pair in P

- k-tuples (c_1, c_2, … , c_k) and (c_1', c_2', … , c_k') are ε –near (complete, separated) if all pairs (c_i, c_j) and (c_i', c_j') are ε –near

Approximate k-clustering

- Thm: Let (c_1, c_2, … , c_k) and (c_1', c_2', …, c_k') be two k-tuples in R^d that are ε-near, $\varepsilon \leq 1/9$. Let $\Pi = \Pi_{Vor}(c_1, c_2, \ldots, c_k)$ and $\Pi' = \Pi_{Vor}(c_1', c_2', \ldots, c_k')$, then $cost(\Pi') \leq (1+6\varepsilon)\, cost(\Pi)$
- Pf:



$$||x-c_1|| \leq ||x-c_1'||+\epsilon\delta \leq ||x-c_2'||+\epsilon\delta \leq ||x-c_2||+2\epsilon\delta$$

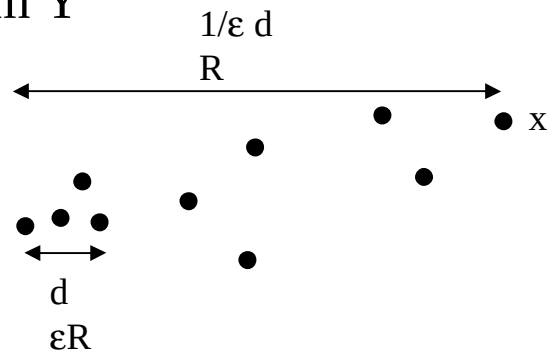
$$||x-c_2|| \geq ||x-c_2'|| - \epsilon\delta \geq \frac{1}{2}\delta - \epsilon\delta$$

$$\delta \leq \frac{2}{1-2\epsilon}||x-c_2|| \leq 3||x-c_2||$$

$$cost(S_1', c_1')+cost(S_2', c_2') \leq cost(S_1', c_1)+cost(S_2', c_2) \leq (1+6\epsilon)(cost(S_1, c_1)+cost(S_2, c_2))$$

Approximate k-clustering

- Instead of looking at all k-tuples in C, we only need to look at ε-complete set.
  - Still too many for near-linear algorithm
- Look at ε –well spread tuples instead, i.e. no subset is 1/ε isolated
  - $Y \subset X$ is 1/ε isolated if there is $x \in X/Y$ such that that is a distance 1/ε diam(Y) from Y



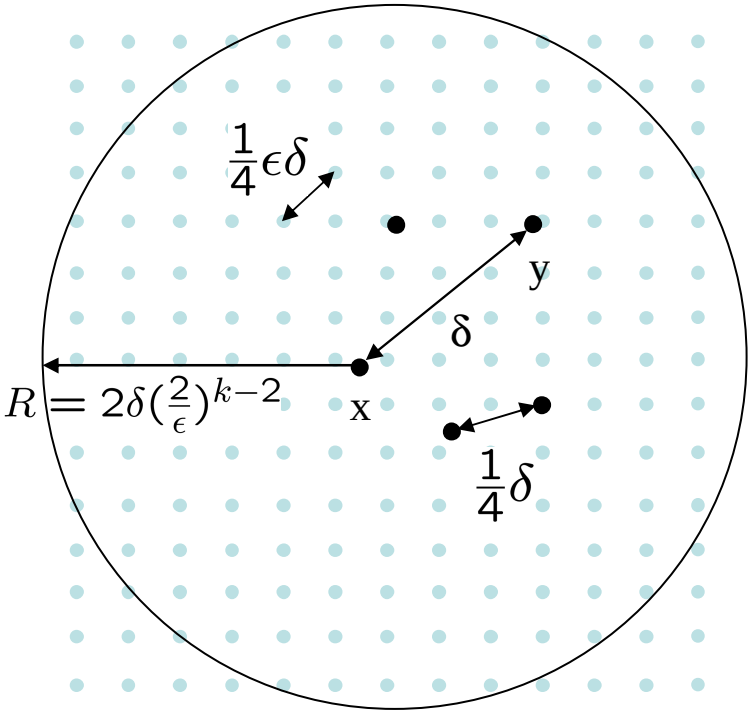
  - If X is ε –well spread, then $\text{diam}(X) \leq (2/\varepsilon)^{k-1}\delta$

Building Cluster Centers

- Let $C \subset R^d$ be an m-point set. Then we can compute the set C' of k-tuples s.t.
  - For any ε –well spread k-tuple in C, there is a tuple in C' that is ε –near it
  - $|C|=O(m\varepsilon^{-k^2d})$
  - Each k-tuple in C' is $\varepsilon/2$ –well spread
  - At least one point in each k-tuple in C' belongs to C
  - There are no more than O(1) k-tuples of C' lying near any given k-tuple in R^d
  - The minimum and maximum distance of points in each k-tuple in C' are bounded by the constant multiples of minimum and maximum distance in C
- The running time is $O(m \log m + m \varepsilon^{-k^2d})$

Building Cluster Centers

- Generate a set of ε/2 –complete set of pairs $P \subset C \times C$
  - There are $O(m\varepsilon^{-d})$ pairs, running time $O(m \log m + m\varepsilon^{-d})$
  - Each pair will be a basis for an ε/2 –well spread k-tuple

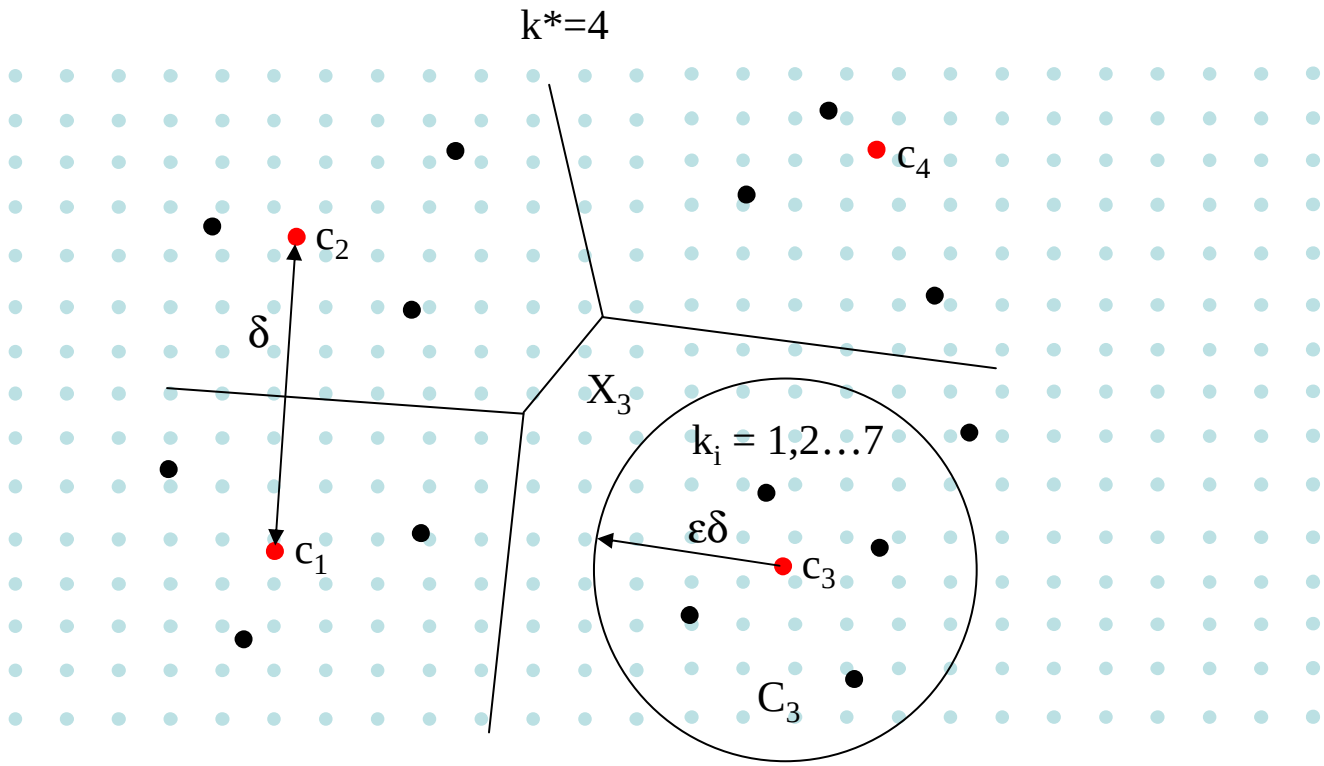


Algorithm

- If k=1, return X
- For k*=2,3…,k generate the set C* of k*-tuples for C
- For each $(c_1, c_2,\ldots,c_{k^*}) \in C^*$, let $(X_1, X_2,\ldots,X_{k^*})=\Pi_{Vor}(c_1, c_2, \ldots,c_{k^*})$
- Let C_i be the points lying in $\varepsilon\delta$ –neighborhood of c_i
  - δ is the smallest pairwise distance among $c_1,\ldots,c_k$
- For i=1…k*, call the algorithm recursively on X_i and C_i
  - vary number of clusters from 1 to k-k*+1
  - find the combination $k_1+k_2+\ldots+k_{k^*}=k$ with the smallest cost
- For each k* tuple in C* with all C_i non-empty, output one with smallest cost

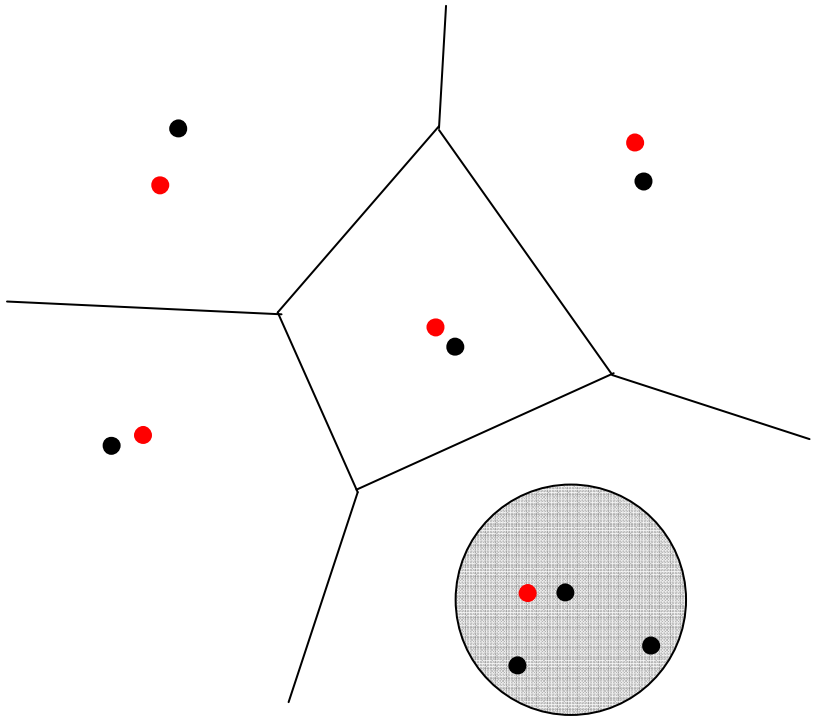
Algorithm

k=10



Correctness

- For any k-tuple $(c_1,\ldots,c_k)$, the algorithm generates a tuple that is ε-near



Running Time

- All range queries are done by approximate range searching in time O(log n)
- Approximate Voronoi partitioning can be done in time $O(\log n + \varepsilon^{-2(d-1)})$
- Running time:

$$
\begin{aligned}
O(|\mathcal{T}_A|(\log n + \varepsilon^{-2(d-1)} + \varepsilon^{-k^2 d})) &= O(|C|(\log |C|)^{k-1}\varepsilon^{-(k-1)^2 d}(\log n + \varepsilon^{-k^2 d})) \\
&= O(n\varepsilon^{-d}(\log n)^k \varepsilon^{-(k-1)^2 d - k^2 d}) \\
&= O(n(\log n)^k \varepsilon^{-2k^2 d}).
\end{aligned}
$$