

# A Survey on Exact and Approximation Algorithms for Clustering

Albert Yu

November 10, 2008

Given a set of point  $P$  in  $\mathbb{R}^d$ , a clustering problem is to partition  $P$  into  $k$  subsets  $\{P_1, P_2, \dots, P_k\}$  in such a way that a given objective function is minimized. The most studied cost functions for a cluster,  $\mu(P_i)$ , are maximum or average radius of  $P_i$ , maximum diameter of  $P_i$ , and maximum width of  $P_i$ . The overall objective function is  $\bigoplus \mu(P_i)$ , where  $\bigoplus$  is typically the  $L_p$ -norm operator. The most common operators are the  $L_1$  (sum) and  $L_\infty$  (max) norms.

This project consists of two parts: The first part is a survey on the min max and min sum clustering problems. This includes 1-center,  $k$ -center,  $k$ -line center,  $k$ -slab cover,  $k$ -median, facility location, and  $k$ -clustering that minimizes the sum of cluster diameters. In the second part, we will study the clustering problem which objective is to minimize the sum of volume of  $k$  axis-aligned rectangles subject to the constraints that all points are covered by at least one rectangle.

## 1 Previous work

### 1.1 MIN MAX Functions

#### 1.1.1 1-center

Given a set of points  $P = \{p_1, p_2, \dots, p_n\} \subset \mathbb{R}^d$ , the 1-center problem is to determine the minimum radius ball enclosing  $P$ . For  $d = 2$ , the 1-center problem can be exactly solved in linear time using the prune-and-search paradigm [13]. This approach extends to higher dimensions with the  $O(d^d n)$  running time for any fixed  $d$  [9]. By the triangle inequality, there exists a simple 2-approximation algorithm, which returns the distance between an arbitrary point and its furthest neighbor. A small coresset,  $Q \subseteq P$ , can also be computed, such that  $(1 + \epsilon)$ -expansion of the radius of  $MEB(Q)$  encloses the entire point set  $P$ . Badoiu et al. [6] provided an incremental algorithm that generates a coresset of size  $O(1/\epsilon^2)$ . At the  $i^{th}$  iteration, their algorithm computes the minimum enclosing ball for the core set  $C_i$ . If the  $(1 + \epsilon)$  expansion of the ball does not cover  $P$ , the point  $p \in P$  that is furthest from the center of the ball is inserted into the core set and the next iteration proceeds with the updated core set  $C_{i+1}$ . Their algorithm is guaranteed to terminate within  $O(1/\epsilon^2)$  iterations and the total running time is  $O(dn/\epsilon^2 + (1/\epsilon)^{10} \log(1/\epsilon))$ . Badoiu and Clarkson [5] later improved the number of required iterations to at most  $2/\epsilon$ . The coresset of size  $O(1/\epsilon^2)$  can be computed in  $O(dn/\epsilon + 1/\epsilon^5)$  time with a simple gradient-descent algorithm.

### 1.1.2 $k$ -center

Given a set of points  $P = \{p_1, p_2, \dots, p_n\} \subset \mathbb{R}^d$ , the  $k$ -center problem is to determine the minimum radius  $r$  such that  $k$  congruent balls of radius  $r$  cover  $P$ . It is NP-hard for  $d \geq 2$  if either  $d$  or  $k$  is part of the input. Agarwal and Procopiuc [1] presented an exact algorithm that solves the  $k$ -center problem in  $\mathbb{R}^2$  in  $O(n\sqrt{k})$  time, under any  $L_p$ -metric. Their algorithm uses dynamic programming and is based on the observation that there exists a set of horizontal lines dividing the plane into strips such that each strip has height at most  $\sqrt{k} + 2$  and each line intersects at most  $\sqrt{k}$  squares of the optimal cover. This implies that the problem can be solved recursively: If the height of the strip is greater than  $\sqrt{k} + 2$ , divide a strip into two smaller strips, and recursively solve the problem on each of the smaller strips. It is not hard to handle the squares that intersect the strip boundary. If the height of a strip is at most  $\sqrt{k} + 2$ , the optimal cover on the strip can be computed using the modified version of the algorithm by Gonzalez [15]. Their algorithm also extends to higher dimensions, and the running time for solving the  $k$ -center problem in  $\mathbb{R}^d$  is  $n^{O(k^{1-1/d})}$ .

Using Gonzalez's greedy algorithm [16], a 2-approximation solution can be computed: At each iteration  $i < k$ , a point that is furthest away from all existing centers is set to be the new center. Badoiu et al. [6] presented an algorithm that computes a  $(1 + \epsilon)$ -approximate  $k$ -center for  $P$  in  $2^{O((k \log k)/\epsilon^2)} dn$  time. Their algorithm runs in  $O(k/\epsilon^2)$  iterations; In each iteration, a point  $p \in P$  that is furthest away from  $k$  minimum enclosing balls is selected and a guessing oracle decides to put  $p$  into one of the  $k$  sets. The oracle can be simulated by exhaustively enumerating all possible guesses, which requires repeating the algorithm  $k^{O(k/\epsilon^2)}$  times. Agarwal and Procopiuc [1] presented a grid-based  $(1 + \epsilon)$ -approximation algorithm with running time  $O(n \log k) + (k/\epsilon)^{O(k^{1-1/d})}$ : The input of the algorithm is a set of grid points  $Q$ , such that for each  $v \in Q$ , at least one of the grid cells adjacent to  $v$  contains a point of  $P$ .

**Definition 1.1**  $Q \subseteq P$  is called an  $(\epsilon, k)$ -certificate for  $P$  if for any set  $I$  of  $k$  intervals that covers  $Q$ , an  $\epsilon$ -expansion of  $I$  results in a cover for  $P$ .

Agarwal et al. [2] proved that a multiplicative  $\epsilon$ -coreset exists for  $k$ -center in  $\mathbb{R}^1$ . Let  $I$  be the interval spanned by  $P$ .  $I$  is first divided into  $k$  intervals of equal length. Let  $a_0 < a_1 < \dots < a_k$  be the endpoints of the intervals. For every endpoint  $a_i$ ,  $0 < i < k$ , an  $(\epsilon, j)$ -certificate is computed for  $P \cap [a_0, a_i]$  and a  $(\epsilon, k - j)$ -certificate for  $P \cap [a_i, a_k]$ , where  $0 < j < k$ . Each interval is further divided into  $O(1/\epsilon)$  intervals, such that  $I$  contains  $O(k/\epsilon)$  endpoints. There are  $O(k^2/\epsilon^2)$  possible intervals defined by these endpoints and for each such interval  $I'$ , a  $(\epsilon, k - 1)$ -certificate is computed for points in  $P$  not covering by  $I'$ . The union of these certificates is an  $(\epsilon, k)$ -certificate for  $P$  and its size is  $O(k/\epsilon)^{O(k)}$ . Har-Peled [17] subsequently proved the existence of a multiplicative  $\epsilon$ -coreset of size  $O(k!/ \epsilon^{dk})$  for  $k$ -center in  $\mathbb{R}^d$ .

### 1.1.3 $k$ -line center

A cylinder in  $\mathbb{R}^d$  is the set of points with a distance of  $r > 0$  from a specified line  $l$ . Given a set of points  $P = \{p_1, p_2, \dots, p_n\} \subset \mathbb{R}^d$ , the  $k$ -line center problem is to determine the minimum radius  $r$  such that  $P$  can be covered by  $k$  cylinders of radius at most  $r$ . Agarwal et al. [2] proved that there exists an additive  $\epsilon$ -coreset of size  $k^{O(k)}/\epsilon^{O(d+k)}$  for the  $k$ -line-center problem. For every cylinder  $C(r, l)$ , a grid of size  $\epsilon$  is drawn in the  $(d - 1)$ -dimensional ball forming the base of  $C$  and  $O(1/\epsilon^{d-1})$  lines parallel to  $l$  are drawn from the grid points. Each point  $p$  is projected to the

nearest line which is within distance  $\epsilon cr$  for some constant  $c$ . Let  $P'$  be the resulting projection of  $P$ . Let  $P'(l) \subseteq P'$  be the points on the line  $l$ . An  $(\delta, k)$ -certificate,  $Q'(l)$ , is computed for  $P'(l)$  for each  $l$  as described in the previous section. By setting  $\delta$  appropriately, The original points in  $P$  corresponding to  $Q' = \bigcup_l Q'(l)$  is an  $\epsilon$ -coreset for  $P$ . Although the proof of the existence of an additive coreset is non-constructive, they also presented the  $O(n \log n)$  expected time algorithm that uses sampling and iterated reweighting techniques for computing the cylinders.

Har-Peled [17] proved that there are no multiplicative coresets for  $k$ -line-center in  $\mathbb{R}^d$ , where  $d \geq 2$ . Define a point set  $P = \{(1/2^i, 2^i) | i = 1, 2, \dots, n\} \in \mathbb{R}^2$ . The goal is to cover  $P$  with 2 strips. Assume  $p(i) = (1/2^i, 2^i)$  is not contained in a  $(1/2)$ -coreset,  $Q$ . The point set  $Q \cap \{p(j) | 1 \leq j < i\}$ , can be covered a horizontal strip of width at most  $2^{i-1}$  and the point set  $Q \cap \{p(j) | i < j \leq n\}$ , can be covered a vertical strip of width at most  $1/2^{i+1}$ . However, expanding the width of both strips by a factor of 1.5 cannot cover  $p(i)$ . Hence,  $Q$  must contain the points  $p(2), \dots, p(n-1)$ .

#### 1.1.4 $k$ -slab cover

A *slab* is defined as the region enclosed between two parallel hyperplanes and the width of the slab is the distance between the two hyperplanes. Given a set of points  $P = \{p_1, p_2, \dots, p_n\} \subset \mathbb{R}^d$ , the  $k$ -slab problem is to find  $k$  slabs of width at most  $w$  such that  $P$  is covered by the  $k$  slabs. Even when  $k = 2$ , Har-Peled [17] proved that any additive  $(1/2)$ -coreset of  $P$  contains  $|P| - 2$  points for the 2-slab problem.

Even though no coreset exists, a near linear time algorithm is provided for approximating the minimum width of 2-slab cover of  $P$  in  $\mathbb{R}^3$ . Assume two points, which lie on the center plane of one of the slabs (say  $S_1$ ), are provided. Further assume those two points lie on the  $x$ -axis. The decision problem (whether a 2-slab cover of width  $(1 + \epsilon)w$  exists) can be determined by rotating the slab  $S_1$  of width  $w$ , such that the center plane has an angle  $\alpha$  with the positive direction of the  $z$ -axis, for  $0 \leq \alpha \leq \pi$ . The set of points covered by  $S_1$  can be maintained for each  $\alpha$  using standard sweeping techniques. At each step, the minimum width slab,  $S_2$ , that covers the remaining points lying outside  $S_1$  is computed. Let  $\alpha^*$  be the angle such that the width of  $S_2$  is minimized. The  $(1 + \epsilon)$ -approximation algorithm computes an  $\alpha$  such that the width of  $S_2(\alpha)$  is at most  $(1 + \epsilon)$  times the width of  $S_2(\alpha^*)$ . To remove the assumption that two points  $p$  and  $q$  lie on the center plane, let  $D_p$  and  $D_q$  denote the spheres of radius  $2w$  around  $p$  and  $q$ , respectively. Let  $N_p$  and  $N_q$  denote the  $(w\epsilon/320)$ -net of size  $O(1/\epsilon^2)$  placing on  $D_p$  and  $D_q$ . If  $\|pq\| \geq \text{diam}(P)/10$ , there must exist two slabs  $S_1$  and  $S_2$  such that  $P$  is covered by  $S_1 \cup S_2$  and the center plane of  $S_1$  passes through a pair of points of  $N_p \cup N_q$ . Given this decision procedure, performing a binary search for the optimal width on the range  $[0, \text{diam}(P)]$  results in a weakly polynomial algorithm. A strongly polynomial algorithm can be achieved by first computing a  $c$ -approximation  $w$  in near linear time, where  $c > 1$  is a constant, and then performing binary search for the optimal width on the range  $[w/c, w]$ . A  $(1 + \epsilon)$  approximation for the 2-slab problem can be computed in  $O(n \log^4 n + n(\log^3 n/\epsilon^8 + 1/\epsilon^{12}) \log 1/\epsilon)$  time.

Assume  $k$  and  $d$  are constants. Edwards and Varadarajan [14] showed that if  $P$  is a subset of integer grid points from the range  $[-\Delta, \Delta]$ , for some  $\Delta > 1$ , there is a coreset  $Q \subseteq P$  of size polynomial in  $(\log \Delta/\epsilon)$ , such that for any  $k$  slabs that cover  $Q$ , their  $\epsilon$ -expansion covers  $P$ .

The coreset construction is inductive:  $(k, \epsilon)$  coresets are built from  $(k - 1, \epsilon)$ . A  $(k - 1, \epsilon)$  coreset  $Q'$  is computed for a small number of appropriately chosen subsets of  $P$ . For any set of  $k$

stabs, a subset of  $P$  contained by the  $k^{\text{th}}$  slab can be approximately identified by the points of  $Q'$  contained by the  $k^{\text{th}}$  slab. A  $(k-1, \epsilon)$  coreset is then computed for the remainder of  $P$ . If no point of  $Q$  is covered by the  $k^{\text{th}}$  slab, the  $\epsilon$ -expansion of the first  $k-1$  stabs can cover  $P$ . The coreset construction uses the following proposition:

**Proposition 1.2** *There exists a constant  $c_d > 0$ , depending only on the dimension  $d$ , such that for any subset  $V \subseteq D$  and point  $q \in D$ ,  $\text{dist}(q, \text{Aff}(V))$  is either 0 or a number in the range  $[c_d/\Delta^d, 4d\Delta]$ .*

By Proposition 1.2, points can be partitioned into  $\log \Delta$  sets, where the  $i^{\text{th}}$  group consists of points that at least  $c_d 2^{i-1}/\Delta^d$  and at most  $c_d 2^i/\Delta^d$  away from an affine subspace. Their algorithm runs in  $d+1$  levels: At each level,  $(k-1, \epsilon)$ -coresets are computed and each point of the coresets may be covered by the  $k^{\text{th}}$  slab. Let  $\text{Aff}(V)$  is the affine space for the current level. For each  $q \in Q'$ , where  $\text{dist}(q, \text{Aff}(V)) > 0$ , the next level is proceeded with the affine space,  $\text{Aff}(V \cup \{q\})$ .  $Q$  can be computed in  $n(\log \Delta/\epsilon)^{f(d,k)}$  time, where  $f(d, k)$  is a function of only  $d$  and  $k$ .

## 1.2 MIN SUM Functions

### 1.2.1 $k$ -median

Given a set of points  $P = \{p_1, p_2, \dots, p_n\} \subset \mathbb{R}^d$ , the  $k$ -center problem is to compute  $k$  points in  $\mathbb{R}^d$  such that the sum of distances from points in  $P$  to their nearest medians is minimized.

**Definition 1.3** ( $(k, \epsilon)$ -coreset for  $k$ -median) *Given a weighted point set  $P \subseteq \mathbb{R}^d$ ,  $S \subseteq \mathbb{R}^d$  is a  $(k, \epsilon)$ -coreset of  $P$  if for any set  $C$  of  $k$  points in  $\mathbb{R}^d$ ,  $(1 - \epsilon)V_C(P) \leq V_C(S) \leq (1 + \epsilon)V_V(P)$ , where  $V_C(P) = \sum_{p \in P} w_p * d(p, C)$ .*

Badoiu et al. [6] presented a  $(1 + \epsilon)$ -approximation algorithms for  $P$  in  $\mathbb{R}^d$ . Using random sampling techniques, their algorithm returns a  $(1 + \epsilon)$ -approximate  $k$ -median in  $2^{(k/\epsilon)^{O(1)}} d^{O(1)} n \log^{O(k)} n$  expected time with high probability.

Har-Peled and Mazumdar [19] showed that a coreset of size  $O(k\epsilon^{-d} \log n)$  can be computed in time  $O(n + k^5 \log^9 n)$ . Their algorithm first computes a set  $A = \{x_1, \dots, x_m\}$  such that  $V_A(P) \leq c * V_{opt}(P)$ , where  $c$  is a constant. Let  $P_i \subseteq P$  denote the points of  $P$  which nearest neighbor in  $A$  is  $x_i$ . An appropriate exponential grid is drawn around for each  $x_i \in A$ . For each grid cell that contains at least one point of  $P_i$ , an arbitrary point of  $P_i$  is picked from the grid cell as a representative point for the coreset. The weight of the chosen point is reset to the sum of the weights of points of  $P_i$  in that grid cell. The size of the coreset is  $O(|A|\epsilon^{-d} \log n)$ . A set  $A$  of size  $k$  can be found as follow: Random sampling techniques are used to get a set  $A$  of size  $O(k \log^3 n)$  such that  $V_A(P) \leq V_{opt}(P)$ . Then a constant factor coreset of size  $O(k \log^4 n)$  is constructed as described above. The constant factor approximation local search algorithm by Arya et al. [4] is then applied to the coreset and the algorithm returns a set  $A$  of size  $k$ .

Har-Peled and Kushal [18] showed that a coreset of size  $O(k^2/\epsilon^d)$ , which is independent of  $n$ , can be constructed for the  $k$ -median problem. When  $d = 1$ , the coreset of size  $O(k/\epsilon)$  can be constructed by breaking the point set into smaller sets and using the mean point of every subset as the representative for the coreset. When  $d > 1$ , a set  $A = \{x_1, \dots, x_k\}$  of size  $k$  is computed as described in the previous paragraph. Let  $P_i \subseteq P$  denote the points of  $P$  which nearest neighbor in  $A$  is  $x_i$ . For each  $x_i \in A$ , let  $N_i$  denote the  $\epsilon/(3c)$ -net placing on the unit sphere centered at  $x_i$ , where  $c$

is the approximation ratio of  $V_A(P)$  to  $V_{opt}(P)$ . For every  $x \in N_i$ , a line spanning the segment  $xx_i$  is generated. Let  $L_i$  denote the set of lines passing through  $x_i$ . Every point  $p \in P_i$  is projected onto the closest line in  $L_i$  for  $1 \leq i \leq k$ . A coreset is constructed for each line in  $\bigcup_{1 \leq i \leq k} L_i$ . Since there are  $O(\epsilon^{-(d-1)})$  lines for each of the  $k$  groups and each line has  $O(k/\epsilon)$  coreset points, the coreset size is  $O(k^2/\epsilon^d)$ . Their result allows a streaming algorithm to use  $O((k^2/\epsilon^d) \log^{d+1} n)$  space to approximate  $k$ -median clustering. Chen [10] reduced the coreset size to  $O(dk\epsilon^{-2} \log n)$ , which has linear dependence on  $d$ , implying that a streaming algorithm only requires  $O(k^2 d \epsilon^{-2} \log^8 n)$  space.

Given a set of point  $P$ , the goal of the discrete  $k$ -median problem is to choose a set of  $k$  points from  $P$  as medians such that the sum of distances from points in  $P$  to their nearest medians is minimized. For the remaining of the section, we will review some of the LP rounding algorithms for discrete  $k$ -median problem.

Lin and Vitter [23] provided an  $\epsilon$ -approximation algorithm that consists of three phases—LP relaxation, filtering, and rounding. In the LP relaxation phase, the optimal fractional solution is solved by linear programming techniques. In the filtering phase, a neighborhood,  $N(i)$ , is defined for each point  $i$  in such a way that even though every point can only connect to a median within its neighborhood, the total connection cost is still bounded by at most  $(1 + \epsilon)opt$ . This is achieved by defining the neighborhood of  $i$  in terms of the expected connection cost of  $i$ , namely  $N(i) = \{j : c_{ij} \leq (1 + \epsilon)C_i\}$ , where  $C_i = \sum_{j \in V} c_{ij} \hat{x}_{ij}$  is the average connection cost of  $i$ . By ensuring that each point  $i$  will only be connected to a median within its neighborhood, the problem can be solved as a set cover problem. Define a set  $S_j = \{i : j \in N(i)\}$  for each  $j$ ; the goal is to select a minimum number of sets such that all the vertices are covered. Therefore, a greedy set cover algorithm can be applied in the rounding phase. The number of medians is bound by  $(\ln n + 1)(1 + 1/\epsilon)k$ .

For the metric  $k$ -median problem, Lin and Vitter [22] further improved the result such that at most  $(1 + 1/\epsilon)k$  medians are used while the total connection cost is bound by a factor of  $2(1 + \epsilon)$ . The main idea is to group a set of points into clusters and use the triangle inequality to bound the connection cost within each cluster. They defined a cluster  $\xi(i)$  to be  $\{j | j \in N(i) \text{ or } N(i) \cap N(j) \neq \emptyset\}$ . That is, either  $j$  is in the neighborhood of  $i$  or their neighborhoods share a common point. Their algorithm repeatedly selects a cluster  $\xi(i)$ , opens a median at  $i$ , and then assigns all the points in the cluster to  $i$ . This process ends when all the points are connected to a median. The key observation is that if  $j' \in \xi(j)$ , the maximum distance between  $j$  and  $j'$  is bounded by  $2(1 + \epsilon) \max\{C_j, C_{j'}\}$  by the triangle inequality. Therefore, by selecting clusters  $\xi(j)$  in ascending order of their average connection costs, the connection cost of  $j' \in \xi(j)$  to the median  $j$  is at most  $2(1 + \epsilon)C_{j'}$ . Thus, the sum of connection costs of vertices to their nearest medians is less than or equal to  $2(1 + \epsilon) \sum_{j \in V} C_j \leq 2(1 + \epsilon) * opt$ . However, their solution is not feasible because more than  $k$  medians are open.

Charikar et. al. [7] used the filtering and rounding techniques to get a constant factor approximation for the metric  $k$ -median problem while using at most  $k$  medians. They consider a generalized version of the  $k$ -median problem, where each point  $j$  has a demand  $d_j$ . Similar to the algorithm mentioned above, the optimal fractional solution is first solved and the average connection cost  $C_j$  is computed for each  $j$ . The main idea of their approach is to first modify the instance by consolidating nearby demands into demands at one single point and nearby fractional medians into one single fractional median, such that after the consolidation procedures, each point will be assigned to a nearby fractional median with probability at least  $1/2$ . Then the fractional solution is converted into a  $\{1/2, 1\}$ -solution, which is then rounded into an integral one solution. The feasible integer solution to the modified instance can then be converted back to a feasible integer solution for the

original instance without increasing the cost too much.

### 1.2.2 Facility Location

Given a set of clients  $D$  and a set of facilities  $F$ , the goal of the uncapacitated facility location problem is to assign each client to an open facility such that the sum of the total connection costs and the total facility costs is minimized. Shmoys et al.[25] provided a constant approximation algorithm which uses two rounding steps. The author first applied the filtering and rounding techniques to enforce the closeness property that each client is only fractionally assigned to nearby facilities. The enforcement increases the expected total facility costs and connection costs by a factor of  $1/\alpha$  and  $1/(1 - \alpha)$ , respectively. In the second rounding step, the author converted a feasible fractional solution into a feasible integer solution without increasing the total facility costs. A cluster  $\xi(j)$  as  $\{i : i \in N(j) \text{ or } N(i) \cap N(j) \neq \emptyset\}$ . is chosen at each iteration and exactly one facility (the one with the minimum facility cost) is open within the cluster. All the clients in the cluster are assigned to the open facility. The procedure is repeated until all the clients are assigned to some open facilities. By the triangle inequality, the connection cost is increased by at most a factor of 3. The approximation ratio is within  $\max\{1/\alpha, 3/(1 - \alpha)\} = 4$  when  $\alpha$  is set to  $1/4$ . By choosing  $\alpha$  at random, the authors improved the approximation guarantee to be 3.16.

Chudak [11] further improved the approximation ratio to  $1 + 2/e$  for the uncapacitated facility location problem. One drawback of Shmoys et al's approach is that the connection cost of a client  $j$  has an approximation factor of at least 3 because even though  $j$  is assigned to an open facility  $i$  in the same cluster,  $i$  may not be in the neighborhood of  $j$ ,  $N(j)$ . To address this problem, Chudak presented a randomized rounding algorithm: When a cluster  $\xi(i)$  is chosen, every facility locations in the cluster has a chance to be open via randomized rounding. Their approach guarantees that for each client  $j \in \xi(i)$ , a facility location will be open within the neighborhood of  $j$  with high probability  $1 - \exp(-1)$ . Therefore,  $j$  will connect to an open facility that is not in its neighborhood with probability at most  $\exp(-1)$ .

One drawback of LP rounding algorithms is that they have to use a linear programming algorithm as a subroutine. Jain and Vazirani [21] developed a combinatorial algorithm that achieves an approximation guarantee of 3 based on the primal dual technique. According to their algorithm, the dual variable of each point is kept raising until the point is connected to an open facility. The primal and dual variables are maintained in such a way that the complementary slackness conditions are satisfied.

### 1.2.3 Minimizing sum of diameters

The diameter of cluster  $C$  is defined as the maximum distance between any two points of  $C$ . Given a set of points  $P = \{p_1, p_2, \dots, p_n\} \subset R^d$ , the goal is to partition the points into  $k$  clusters such that the sum of cluster diameters is minimized. Doddi et al [12] proved that it is NP-hard to obtain an approximation factor  $2 - \epsilon$  for any  $\epsilon > 0$ . For non-metric, it is NP-hard to obtain any approximation factor even when  $k = 3$ . On the positive side, the problem can be solved efficiently for  $k = 2$  [20, 24]. Doddi et al [12] presented an  $O(\log(n/k))$ -approximation algorithm using at most  $10k$  clusters in metric spaces. The algorithm is based on the observation that if two clusters  $C_1$  and  $C_2$  are not disjoint, the diameter of  $C_1 \cup C_2$  is at most the sum of the diameters of  $C_1$  and  $C_2$ . The algorithm maintains a set  $D$  of clusters which cover  $P$ . Initially,  $D$  consists of  $n$  singleton clusters. At each iteration, a point set  $P'$  is constructed by selecting an arbitrary point from each

cluster. The clustering problem for  $P'$  is transformed to the weighted set cover problem, and a cover  $D'$  of size  $O(k \log(|D|/k))$  is returned. Since each cluster of  $D$  must intersect with a set (cluster) of  $D'$ , the two clusters can be merged together without increasing the sum of cluster diameters. The algorithm stops when the number of clusters is at most  $10k$ .

Charikar and Panigraphy [8] presented a primal-dual algorithm that obtains a constant factor approximation using at most  $k$  clusters. They also provided an incremental algorithm that maintains a constant factor approximate solution with a constant factor blowup in the number of clusters. They actually deal with the min radii formulation in the paper since  $\epsilon$ -approximation algorithms for minimizing the sum of cluster diameters give  $2\epsilon$ -approximation algorithms for minimizing the sum of cluster radii in metric spaces.

## 2 Minimizing sum of volume

Given a set  $P = \{p_1, p_2, \dots, p_n\} \in \mathbb{R}^d$ , the goal is to find  $k$  axis-aligned rectangles  $R = \{r_1, \dots, r_k\}$  that cover  $P$ , such that the sum of the volume of the rectangles is minimized. Surprisingly little is known about this objective function. For  $k = 2$ , Arkin et. al. [3] presented a  $(1 + \epsilon)$ -approximation algorithm that runs in  $O(n \min\{\log n, 1/\epsilon\} + n/\epsilon^2)$  time in  $\mathbb{R}^3$ . Their result can be generalized to higher dimension with running time  $O(d^2 n \min\{\log n, 1/\epsilon\} + \epsilon^{-\lceil d/2 \rceil} d^2 n^{\lfloor d/2 \rfloor} (d C \lceil d/2 \rceil))$ .

### 2.1 No coresets

**Definition 2.1**  $Q \subseteq P$  is a multiplicative coreset for  $P$  if for any set  $R$  of  $k$  rectangles that covers  $Q$ , an  $\epsilon$ -expansion (increasing each side of a rectangle by a factor of  $1 + \epsilon$ ) of  $R$  covers  $P$ .

No multiplicative coreset of size less than  $o(n)$  exists if the points of  $P$  can have real coordinates. The proof is the same as the proof of no multiplicative coresets for  $k$ -line center problem, except that the coreset is for the problem of covering  $P$  with two rectangles.

These are some possible problems for further research:

1. If the points comes from an integer grid  $[\Delta]^d$ , can we find a coreset whose size is logarithmic to  $\Delta$ ?
2. Although a multiplicative coreset does not exist, can we develop other techniques to efficiently approximate the solution?
3. Does there exist a coreset that gives good approximation if we have a different definition of  $\epsilon$ -expansion? For instance, if a rectangle is expanded as the Minkowski sum of the rectangle and a small ball of radius  $r$ , can we efficiently find a small coreset and  $r$ , such that the union of the expanded rectangles cover the original point set and the sum of volume only increases by a factor of  $(1 + \epsilon)$ ?

### 2.2 When the rectangles are fat

When the rectangles are fat, the approximate solution can be computed efficiently. Consider the problem of covering  $P$  with  $k$  squares in  $\mathbb{R}^2$ . We first find the minimum bounding square that covers the entire point set  $P$  and create a quadtree. Zoom in to the level which has  $k(2k + 1) + 1$

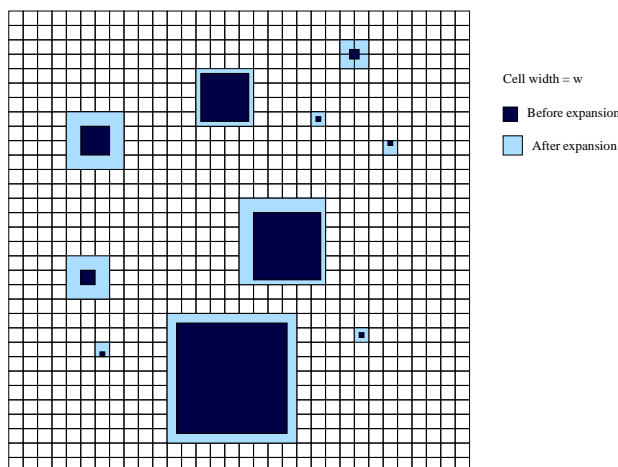


Figure 1: Expansion of squares: Every square is expanded such that every partially covered cell by rectangle  $r$  becomes entirely covered by  $r$ . The  $k$  squares that cover  $Q$  will also cover  $P$  after expansion. The expansion increases the sum of volume of the squares by a factor of at most 7.

non-empty columns. Let  $w$  denote the width of a cell at that level. For each non-empty cell  $C_i$ , an arbitrary point of  $P \cap C_i$  is chosen to be the representative of all the points in  $P \cap C_i$  and it is added to the coreset  $Q$ . First, for any set of  $k$  squares that covers  $Q$ , if every square  $R$  is expanded such that every partially covered cell by rectangle  $R$  becomes entirely covered by  $R$  (see Figure 1), the  $k$  expanded squares will cover the entire set  $P$ . This is because every point  $p \in P$  has a representative in  $Q$ , so the cell containing  $p$  must be at least partially covered. Next, let  $OPT(P)$  and  $OPT(Q)$  denote the minimum sum of volume of  $k$  squares that cover  $P$  and  $Q$ , respectively.  $OPT(Q) \leq OPT(P)$  since  $Q \subseteq P$ . We will show that the expansion increases the sum of the volume of the squares by a factor of at most 7. Observe that if the number of non-empty columns is  $k(2k + 1) + 1$ , then at least 1 square has width at least  $2kw$ , where  $w$  is the width of a cell. Therefore,  $OPT(Q)$  is at least  $4k^2w^2$ . For each square  $R_i$  that cover  $Q$ , let  $l_i$  denote its width. The expansion of  $R_i$  will increase its volume by  $(l_i + 2w)^2 = l_i^2 + 4w^2 + 4l_iw$ . The second term is at most  $OPT(Q)/k$ . The third term is at most  $OPT(Q)/k$  if  $l_i < w$ , and at most  $4 * l_i^2$  if  $l_i \geq w$ .  $\sum_{1 \leq i \leq k} Volume(Expanded(R_i)) = \sum_{1 \leq i \leq k} (l_i^2 + 4w^2 + 4l_iw) = \sum_{1 \leq i \leq k} l_i^2 + \sum_{1 \leq i \leq k} 4w^2 + \sum_{1 \leq i \leq k} 4l_iw \leq OPT(Q) + OPT(Q) + 5OPT(Q) = 7OPT(Q)$ . Note that an  $(1 + \epsilon)$  approximation solution can be achieved if each cell is further divided into  $1/\epsilon^2$  cells.

## References

- [1] P.K. Agarwal and C.M. Procopiuc. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2002), 201-206.
- [2] P.K. Agarwal, C.M. Procopiuc, and K. R. Varadarajan. Approximation algorithms for  $k$ -line center. *Proc. 10th Annu. European Sympos. Algorithms*, 2002, pp. 54-63.

- [3] E.M. Arkin, G. Barequet, and J.S.B. Mitchell. Algorithms for Two-Box Covering. *Proc. 22th Annu. Sympos. on Comput. Geometry*, 2006, pp. 459-467.
- [4] V. Arya, N. Garg, R. Khandekar, K. Munagala, and V. Pandit. Local search heuristic for  $k$ -median and facility location problems. In *Proc. 33rd Annu. ACM Sympos. Theory Comput.*, 2001, 21-29.
- [5] M. Badoiu and K. Clarkson. Smaller core-sets for balls. *Proc. 14th ACM-SIAM Sympos. Discrete Algorithms*, 2003, 801-802.
- [6] M. Badoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. *Proc. 34th Annu. ACM Sympos. Theory Comput.*, 2002, 250-257.
- [7] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the  $k$ -median problem. *J. Comput. Sys. Sci.*, 65(1):129–149, 2002.
- [8] M. Charikar and R. Panigraphy. Clustering to minimize the sum of cluster diameters. *Journal of Computer and Systems Sciences*, Vol.68(2), Ppp. 417-441, 2004.
- [9] B. Chazelle and J. Matousek. On linear-time deterministic algorithms for optimization problems in fixed dimensions. *J.Alg. 21*, 579-597.
- [10] K. Chen. On  $k$ -Median Clustering in High Dimensions. *SODA'06*, 1177-1185.
- [11] F. A. Chudak. Improved Approximation Algorithms for Uncapacitated Facility Location. In *proceedings of the 6th IPCO Conference*, pages 180-194, 1998.
- [12] S.R. Doddi, M.V. Marathe, S.S. Ravi, D.S. Taylor, and P. Widmayer. Approximation algorithms for clustering to minimize the sum of diameters. In *proceedings of the 7th Scandinavian Workshop on Algorithm Theory*, pages 237-250, 2000.
- [13] M.E. Dyer. On a multidimensional search technique and its application to the Euclidean one-center problem. *SIAM J. Comput.* 15, 725-738.
- [14] M. Edwards and K. R. Varadarajan. No Coreset, No Cry: II. *FSTTCS*, 2005: 107-115
- [15] T. Gonzalez. Covering a set of points in multidimensional space. *Inform. Process. Lett.*, 40(1991), 181-188.
- [16] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoret. Comput. Sci.*, 38(1985), 293-306.
- [17] S. Har-Peled. No coreset, no cry. *Proc. 24th Conf. Found. Soft. Tech. Theoret. Comput. Sci.*, 2004.
- [18] S. Har-Peled and A. Kushal. Smaller coresets for  $k$ -median and  $k$ -means clustering. *Proc. 36th Annu. ACM Sympos. Theory Comput.*, 291-300, 2004.
- [19] S. Har-Peled and S. Mazumdar. Coresets for  $k$ -means and  $k$ -median clustering and their applications. *Proc. 36th Annu. ACM Sympos. Theory Comput.*, 2004, 291-300.
- [20] P. Hansen and B. Jaumard. Minimum sum of diameters clustering *Journal of Classification*, Vol. 4, 1987, pp. 215-226.
- [21] K. Jain, and V.V. Vazirani Approximation algorithms for metric facility location and  $k$ -median problems using the primal-dual schema and Lagrangian relaxation. In *Journal of the ACM*, 48 (2001), 274-296.
- [22] Jyh-Han Lin, Jeffrey Scott Vitter Approximation algorithms for geometric median problems. *IPL* 44, 245-249, 1992.

- [23] Jyh-Han Lin, Jeffrey Scott Vitter.  $\epsilon$ -Approximations with Minimum Packing Constraint Violation (Extended Abstract) *STOC* 1992: 771-782.
- [24] C.L. Monma and S. Suri. Partitioning points and graphs to minimize the maximum of the sum of diameters. *Proc. 6th Int. Conf. Theory and Applications of Graphs*, Kalamazoo, Michigan, May 1989.
- [25] D. B. Shmoys, Eva Tardos, and K. Aardal. Approximation algorithms for facility location problems (extended abstract). In *Proceedings of ACM symposium on Theory of computing (STOC)*, May 1997.