A. The Combinatorial Complexity of the Sets Clustering

The following definition of a query space encapsulates all the ingredients required to formally define an optimization problem.

**Definition A.1** (Query space; see Definition 4.2 in (Braverman et al., 2016)). Let \( \mathcal{P} \) be a set called input set. Let \( Q \) be a (possibly infinite) set called query set. Let \( f : \mathcal{P} \times Q \to \mathbb{R} \) be a cost function. The tuple \( (\mathcal{P}, Q, f) \) is called a query space. A sets clustering query space is a query space \( (\tilde{P}, Q, f) \) where \( \tilde{P} \) is an \((n,m)\)-set, \( Q \) is the set \( X_k \), and \( f = \tilde{D} \); see Section 2.

In what follows we define some measure of combinatorial complexity for a query space.

**Definition A.2** (Definition 4.5 in (Braverman et al., 2016)). For a query space \((\mathcal{P}, Q, f)\), a query \( C \in Q \) and \( r \in [0, \infty) \) we define

\[
\text{range}(\mathcal{P}, C, r) = \{ P \in \mathcal{P} | f(P, c) \leq r \}.
\]

Let \( \text{ranges}(\mathcal{P}, Q, f) = \{ \text{range}(\mathcal{P}, C, r) | C \in Q, r \geq 0 \} \), the VC-dimension of \((\mathcal{P}, \text{ranges}(\mathcal{P}, Q, f))\) is the smallest integer \( d' \) such that for every \( \mathcal{H} \subseteq \mathcal{P} \) we have

\[
\left| \{ \text{range}(C, r) | C \in Q, r \in [0, \infty) \} \right| \leq |\mathcal{H}|^{d'}.
\]

The dimension of the query space \((\mathcal{P}, Q, f)\) is the VC-dimension of \((\mathcal{P}, \text{ranges}(\mathcal{P}, Q, f))\).

**Lemma A.3** (Variant of Theorem 8.4, (Anthony & Bartlett, 2009)). Suppose \( h \) is a function from \( \mathbb{R}^d \times \mathbb{R}^n \) to \{0, 1\} and let

\[
H = \{ h(a, x) | a \in \mathbb{R}^d, x \in \mathbb{R}^n \}
\]

be the class determined by \( h \). Suppose that \( h \) can be computed by an algorithm that takes as an input a pair \((a, x)\) \( \in \mathbb{R}^d \times \mathbb{R}^n \) and returns \( h(a, x) \) after no more than \( t \) operations of the following types:

- the arithmetic operations +, -, \times, and / on real numbers,
- jumps conditioned on \( >, \geq, <, \leq, = \), and \( \neq \) comparisons of real numbers, and
- outputs 0 or 1.

Then the VC-dimension of \( H \) is \( O(dt) \).

We now bound the dimension of a query space \((\mathcal{P}, X_k, \tilde{D})\) as in Definition A.2.

**Lemma A.4.** Let \((\mathcal{P}, X_k, \tilde{D})\) be a sets clustering query space; see Definition A.1. Then the dimension \( d' \) of \((\mathcal{P}, X_k, \tilde{D})\) is bounded by \( O(md^2k^2) \).

**Proof.** For \( P \in \mathcal{P}, C \in X_k \), and \( r \in \mathbb{R} \), let \( h_P(C, r) = 1 \) if \( \tilde{D}(P, C) \geq r \) and 0 otherwise. Then we observe that the VC-dimension of the class of functions \( H = \{ h_P : X_k \times \mathbb{R} \to [0, \infty) | P \in \mathcal{P} \} \) in Lemma A.3 is equivalent to the dimension \( d' \) of the given query space. Therefore, we now show that the VC-dimension of \( H \) is bounded by \( O(md^2k^2) \).

Note that it takes \( t = O(md^2) \) arithmetic operations to evaluate \( h_P(C, r) \). Furthermore, any element in \( X_k \times \mathbb{R} \) can be represented as a vector in \( (dk + 1) \)-dimensional space. Hence by Lemma A.3, the VC-dimension of \( H \) is \( O(dk \cdot md^2) = O(md^2k^2) \).

B. Main theorems with full proof

**B.1. Proof of Lemma 4.1**

**Lemma B.1.** Let \( k \geq 1 \) be an integer, \( A, B \subseteq X \) and \( C \subseteq X_k \). If \( \tilde{D}(A \cup B, C) \neq \tilde{D}(B, C) \) then \( \tilde{D}(A \cup B, C) = \tilde{D}(A, C) \).

**Proof.** By definition, \( \tilde{D}(A \cup B, C) = \min \{ \tilde{D}(A, C), \tilde{D}(B, C) \} \). By the assumption of the lemma, \( \tilde{D}(A \cup B, C) \neq \tilde{D}(B, C) \). Therefore, \( \tilde{D}(A \cup B, C) = \tilde{D}(A, C) \).

**Lemma B.2.** Let \( A = \{ a_1, \cdots, a_n \} \subseteq X \) and put \( b \in X \). Let \( B = (A \setminus \{ a_1 \}) \cup \{ b \} = \{ b, a_2, \cdots, a_n \} \subseteq X \). Then for every \( C \in X_k \) we have that

\[
\tilde{D}(A, C) \leq \rho D(B, C) + \tilde{D}(a_1, b) .
\]

**Proof.** By definition, we have that

\[
\begin{align*}
\tilde{D}(A,C) & = \min \left\{ \tilde{D}(a_1,C), \tilde{D}(A \setminus \{ a_1 \}, C) \right\} \\
& \leq \min \left\{ \rho \left( \tilde{D}(a_1,b) + \tilde{D}(b,C) \right), \tilde{D}(A \setminus \{ a_1 \}, C) \right\} \\
& \leq \min \left\{ \rho \left( \tilde{D}(a_1,b) + \tilde{D}(b,C) \right), \rho \left( \tilde{D}(A \setminus \{ a_1 \}, C) + \tilde{D}(a_1,b) \right) \right\} \\
& \leq \rho \min \left\{ \tilde{D}(b,C), \tilde{D}(A \setminus \{ a_1 \}, C) \right\} + \rho \tilde{D}(a_1,b) \\
& = \rho \tilde{D}(B,C) + \rho \tilde{D}(a_1,b),
\end{align*}
\]

where the first inequality is by the weak triangle inequality by Lemma 2.2, and the last derivation is by the definition of \( B \).
output of a call to RECURSIVE-ROBUST-MEDIAN(\(P, k\)); see Algorithm 1. Then, for every \(P \in \mathcal{P}^m\) we have that
\[
\sup_{C \in \mathcal{X}_k} \frac{\hat{D}(P, C)}{D(Q, C)} \leq O(1) \cdot \left( \frac{1}{|\mathcal{P}^m|} \right).
\]

Proof. In what follows, we use the variables and notations from Algorithm 1. Put \(P \in \mathcal{P}^m, i \in [m]\), and consider the \(i\)th iteration of the “for” loop at Line 4 of Algorithm 1. Put \(C \in \mathcal{X}_k\).

Let
\[
\mathcal{P}^{i-1} = \left\{ Q \in \mathcal{P}^{i-1} \mid \hat{D}(T(Q, B^{i-1}), C) = \hat{D}(B^{i-1}, C) \right\}
\]
be the union of sets \(Q \in \mathcal{P}^{i-1}\) whose closest point to the query \(C\) after the projection on \(B^{i-1}\) is one of the points of \(B^{i-1}\). First, we prove that
\[
\frac{\hat{D}(T(P, B^{i-1}), C)}{\sum_{Q \in \mathcal{P}^{i-1}} \hat{D}(T(Q, B^{i-1}), C)} \leq 3\rho^2 \sum_{Q \in \mathcal{P}^{i-1}} \hat{D}(T(Q, B^i), C) + \frac{4\rho}{|\mathcal{P}|}
\]
by the following case analysis: (i) \(|\mathcal{P}^{i-1}| \geq \frac{|\mathcal{P}^{i-1}|}{2}\), i.e., more than half the sets satisfy that their closest point to \(C\) is amongst their projected points onto \(B^{i-1}\), and (ii) Otherwise, i.e., \(|\mathcal{P}^{i-1}| < \frac{|\mathcal{P}^{i-1}|}{2}\).

Case (i): \(|\mathcal{P}^{i-1}| \geq \frac{|\mathcal{P}^{i-1}|}{2}\). By Line 7 we have
\[
\mathcal{P}^{i} \subseteq \mathcal{P}^{i-1} \subseteq \cdots \subseteq \mathcal{P}^0 = \mathcal{P}.
\]
Therefore,
\[
\sum_{Q \in \mathcal{P}^{i-1}} \hat{D}(T(Q, B^{i-1}), C) \geq \sum_{Q \in \mathcal{P}^{i-1}} \hat{D}(T(Q, B^{i-1}), C)
\]
\[
= \sum_{Q \in \mathcal{P}^{i-1}} \hat{D}(B^{i-1}, C) \geq \frac{|\mathcal{P}^{i-1}|}{2} \hat{D}(B^{i-1}, C),
\]
where (5) holds since \(\mathcal{P}^{i-1} \subseteq \mathcal{P}^{i-1}\), the first derivation in (6) is by the definition of \(\mathcal{P}^{i-1}\), and the second derivation in (6) is by the assumption of Case (i). This proves (3) for Case (i) as
\[
\frac{\hat{D}(T(P, B^{i-1}), C)}{\sum_{Q \in \mathcal{P}^{i-1}} \hat{D}(T(Q, B^{i-1}), C)} \leq \frac{\hat{D}(B^{i-1}, C)}{D(Q, C)} \leq \frac{2}{|\mathcal{P}^{i-1}|} \leq \frac{2}{|\mathcal{P}|},
\]
where the first inequality holds since \(B^{i-1} \subseteq T(P, B^{i-1})\) by Definition 2.5, and the second inequality is by (6).

Case (ii): \(|\mathcal{P}^{i-1}| < \frac{|\mathcal{P}^{i-1}|}{2}\). Let \(\gamma = 1/(2k)\). Let \(\hat{P}^{i-1}, b_i\) and \(P^i\) be as defined in Lines 5, 6, and 7 respectively, and identify \(B^{i-1} = \{b_1, \ldots, b_i\}\) for \(i \geq 2\) or \(B^{i-1} = \emptyset\) for \(i = 1\). Let
\[
OPT_i = \min_{C \in \mathcal{X}_k} \sum_{P \in \mathcal{P}^{i-1}, C \neq \mathcal{P}^{i-1}, C'} \hat{D}(\hat{P}, C').
\]
For every \(Q \in \mathcal{P}^{i-1}\), substituting \(A = \text{proj}(Q, B^{i-1})\) and \(B = B^{i-1}\) in Lemma B.1 proves that
\[
\left\{ \begin{array}{c} Q \in \mathcal{P}^{i-1} \rightarrow \hat{D}(\text{proj}(Q, B^{i-1}) \cup B^{i-1}, C) \neq \hat{D}(B^{i-1}, C) \\
\in \mathcal{P}^{i-1} \rightarrow \hat{D}(\text{proj}(Q, B^{i-1}) \cup B^{i-1}, C) = \hat{D}(\text{proj}(Q, B^{i-1}), C) \end{array} \right\}
\]
We now obtain that
\[
\left\{ \begin{array}{c} Q \in \mathcal{P}^{i-1} \rightarrow \hat{D}(T(Q, B^{i-1}), C) = \hat{D}(\text{proj}(Q, B^{i-1}), C) \end{array} \right\}
\]
\[
= \left\{ \begin{array}{c} Q \in \mathcal{P}^{i-1} \rightarrow \hat{D}(\text{proj}(Q, B^{i-1}) \cup B^{i-1}, C) = \hat{D}(\text{proj}(Q, B^{i-1}), C) \end{array} \right\}
\]
\[
= \left\{ \begin{array}{c} Q \in \mathcal{P}^{i-1} \rightarrow \hat{D}(T(Q, B^{i-1}), C) \neq \hat{D}(B^{i-1}, C) \end{array} \right\}
\]
\[
= |\mathcal{P}^{i-1}| \mathcal{P}^{i-1} \geq \frac{|\mathcal{P}^{i-1}|}{2},
\]
where (10) and (12) is by substituting \(P = Q\) and \(B = B^{i-1}\) in Definition 2.5, (11) is by (9), the first derivation in (13) is by the definitions of \(\mathcal{P}^{i-1}\) and \(\mathcal{P}^{i-1}\), and the last inequality is by the assumption of Case (ii).

Recall that by Line 5,
\[
\hat{P}^{i-1} = \{ \text{proj}(Q, B^{i-1}) \mid Q \in \mathcal{P}^{i-1} \},
\]
and let
\[
Z = \left\{ Q \in \mathcal{P}^{i-1} \mid \text{proj}(Q, B^{i-1}) \in \text{closest}(\hat{P}^{i-1}, C) \right\}.
\]
Since \(Z\) contains the \(|Z| \geq \frac{|\mathcal{P}^{i-1}|}{2}\) sets \(Q \in \mathcal{P}^{i-1}\) with the smallest \(\hat{D}(\text{proj}(Q, B^{i-1}), C)\), for any set \(Z' \subseteq \mathcal{P}^{i-1}\) such that \(|Z'| \geq \frac{|\mathcal{P}^{i-1}|}{2}\), we have
\[
\sum_{Q \in Z} \hat{D}(\text{proj}(Q, B^{i-1}), C) \leq \sum_{Q \in Z'} \hat{D}(\text{proj}(Q, B^{i-1}), C).
\]
By the assumption of Case (ii),
\[ |P^{i-1} \setminus \overline{P}^{i-1}| \geq \frac{|P^{i-1}|}{2}, \tag{15} \]
and by the definition of \( Z \), we have
\[ \{ \text{proj}(Q, B^{i-1}) \mid Q \in Z \} = \text{closest}(\hat{P}^{i-1}, C, 1/2). \tag{16} \]
Therefore,
\[
\sum_{Q \in \text{closest}(\hat{P}^{i-1}, C, 1/2)} \hat{D}(\hat{Q}, C) \\
= \sum_{Q \in Z} \hat{D}(\text{proj}(Q, B^{i-1}), C) \\
\leq \sum_{Q \in P^{i-1} \setminus \overline{P}^{i-1}} \hat{D}(\text{proj}(Q, B^{i-1}), C), \tag{17}
\]
where the first derivation is by (16) and the last derivation is by substituting \( Z' = P^{i-1} \setminus \overline{P}^{i-1} \) in (14). By the definitions of \( P^{i-1} \) and \( \overline{P}^{i-1} \), for every \( Q \in P^{i-1} \setminus \overline{P}^{i-1} \), we have
\[ \hat{D}(T(Q, B^{i-1}), C) = \hat{D}(\text{proj}(Q, B^{i-1}), C). \tag{18} \]
Hence,
\[
\text{OPT}_i \leq \sum_{Q \in \text{closest}(P^{i-1}, C, 1/2)} \hat{D}(\hat{Q}, C) \\
\leq \sum_{Q \in P^{i-1} \setminus \overline{P}^{i-1}} \hat{D}(\text{proj}(Q, B^{i-1}), C) \\
= \sum_{Q \in P^{i-1} \setminus \overline{P}^{i-1}} \hat{D}(T(Q, B^{i-1}), C) \\
\leq \sum_{Q \in P^{i-1}} \hat{D}(T(Q, B^{i-1}), C), \tag{20}
\]
where (19) holds by the definition of \( \text{OPT}_i \), (20) is by (17), and (21) is by (18).

Recall that \( P \in P^m \), identify
\[ \text{closepairs}(P, B^m) = \left\{ (\hat{p}_1, \hat{b}_1), \ldots, (\hat{p}_m, \hat{b}_m) \right\}, \]
as in Definition 2.5 (i). Also by Definition 2.5, for every \( i \in [m] \) we have
\[
\hat{D}(\text{proj}(P, B^{i-1}), \hat{b}_i) \\
= \hat{D}(P \setminus \{ \hat{p}_1, \ldots, \hat{p}_{i-1} \}, \hat{b}_i) \\
= \hat{D}(\hat{p}_i, \hat{b}_i). \tag{23}
\]
Since \( P \in \mathcal{P}^{i} \) and \( \gamma = \frac{1}{2\epsilon} \), we have by Line 7 that
\[ \text{proj}(P, B^{i-1}) \in \text{closest}(\hat{P}^{i-1}, \{ b^i \}, (1 - \tau)\gamma / 2). \tag{24} \]
Observe that in the definition of \( \text{OPT}_i \) in (8), the largest cluster in every set \( C' \) of \( k \) centers contains at least \( \frac{|P^{i-1}|}{2k} \) points by the Pigeonhole Principle. Therefore, since the cost of the closest \( (1 - \tau)\gamma |\hat{P}^{i-1}| \) sets for \( b^i \) is a 2-approximation for the optimal set of \( \gamma |\hat{P}^{i-1}| \) points, we have
\[
\sum_{Q \in \text{closest}(\hat{P}^{i-1}, \{ b^i \}, (1 - \tau)\gamma)} \hat{D}(Q, \hat{b}_i) \\
\leq 2 \min_{\{ b \} \in A} \sum_{Q \in \text{closest}(\hat{P}^{i-1}, \{ b \}, \gamma)} \hat{D}(Q, b) \tag{25}
\]
Therefore,
\[
\hat{D}(\hat{p}_i, \hat{b}_i) = \hat{D}(\text{proj}(P, B^{i-1}), \hat{b}_i) \\
\leq 2 \cdot \text{OPT}_i. \tag{26}
\]
where (26) is by (23), (27) is by combining Markov’s Inequality with (24), (28) follows since \( |P^i| = \frac{(1 - \tau)\gamma}{\epsilon} |P^{i-1}| \leq (1 - \tau)\gamma |\hat{P}^{i-1}| \), and (29) is by (25).

Now, since the sets \( T(P, B^{i-1}) \) and \( T(P, B^i) \) differ by at most one point, i.e.,
\[ T(P, B^i) = (T(P, B^{i-1}) \setminus \{ \hat{p}_i \}) \cup \{ b_i \}, \]
by substituting \( A = T(P, B^{i-1}) \), and \( B = T(P, B^i) \) in Lemma B.2, we obtain that
\[ \hat{D}(T(P, B^{i-1}), C) \leq \rho \hat{D}(T(P, B^i), C) + \rho \hat{D}(\hat{p}_i, \hat{b}_i). \tag{30} \]
By the previous inequality we obtain
\[
\sum_{Q \in P^{i-1}} \hat{D}(T(Q, B^{i-1}), C) \leq \rho \sum_{Q \in P^{i-1}} \hat{D}(T(Q, B^i), C) \tag{31}
\]
\[ + \rho \sum_{Q \in P^{i-1}} \hat{D}(\hat{p}_i, \hat{b}_i) \]
We can now apply

(32)

where (32) is by (22), and the first derivation in (33) is by (29).

We now bound the middle term of (31). By identifying closepairs\(Q, B^m = \left\{ (q_1, b_1), \ldots, (q_m, b_m) \right\}\) for every \(Q \in P^i\), we have,

\[
\sum_{Q \in P^i} \tilde{D}(T(Q, B^i), C) \leq \rho \sum_{Q \in P^i} \tilde{D}(T(Q, B^{i-1}), C) + \rho \sum_{Q \in P^i} \tilde{D}(q_i, b_i) \tag{34}
\]

(35) holds similarly to (30), (35) holds similarly to (29) for the set \(P\) instead of \(P, (36)\) holds since \(P^i \subseteq P^{i-1}\) by (4) and (37) is by (22). Thus, by (37), the middle term of (31) is bounded by

\[
\rho \sum_{Q \in P^{i-1}} \tilde{D}(T(P, B^i), C) = \frac{3 \rho^2}{4} \sum_{Q \in P^{i-1}} \tilde{D}(T(Q, B^i), C). \tag{38}
\]

By combining (31), (33) and (38), we get that

\[
\frac{\sum_{Q \in P^{i-1}} \tilde{D}(T(P, B^{i-1}), C)}{\sum_{Q \in P^{i-1}} \tilde{D}(T(Q, B^{i-1}), C)} \leq 3 \rho^2 \left( \frac{\rho}{|P^i|} \right)^2 \tag{39}
\]

Now (3) holds by taking the maximum between the bounds of Case (i) in (7), and the bound of Case (ii) in (39).

We can now apply (3) recursively over every \(i \in [m]\) to obtain that

\[
\sum_{Q \in P} \tilde{D}(P, C) \leq \frac{3 \rho^2}{4} \sum_{Q \in P} \tilde{D}(T(P, B^i), C) \tag{40}
\]

Also, for every \(Q \in P^m\) observe that \(|Q| = |B^m| = m\), hence

\[
T(Q, B^m) = B^m = \left\{ b_1, \ldots, b_m \right\}.
\]

Thus, for every \(Q \in P^m\) and \(C \in \mathcal{X}_k\)

\[
\tilde{D}(T(Q, B^m), C) = \tilde{D}\left( \left\{ b_1, \ldots, b_m \right\}, C \right) \tag{42}
\]

Lemma 4.1 now holds as

\[
\sum_{Q \in P} \tilde{D}(P, C) \leq \frac{3 \rho^2}{4} \sum_{Q \in P} \tilde{D}(T(P, B^i), C) \leq \frac{3 \rho^2}{4} \sum_{Q \in P} \tilde{D}(T(Q, B^i), C) + 4 \rho \sum_{i \in [m]} \frac{3 \rho^2}{|P^i|} \tag{43}
\]

(44) holds since \(|P^m| \leq |P^i|\) for every \(i \in [m]\), (45) holds by summing the geometric sequence, and inequalities (46) and (47) hold since \(\rho \geq 1\).

\[\square\]

B.2. Proof of Theorem 4.2

Theorem 4.2. Let \(P\) be an \((n, m)\)-set, \(k \geq 1\) be an integer, \((\mathcal{X}, \tilde{D})\) be as in Definition 2.1, and \(\varepsilon, \delta \in (0, 1)\). Let \((S, v)\) be the output of a call to CORESET\((P, k, \varepsilon, \delta)\). Then

(i) \(|S| \in O\left( \frac{m d \log n}{\varepsilon} \right)^2 k^{m+4} \).

(ii) With probability at least \(1 - \delta\), \((S, v)\) is an \(\varepsilon\)-coreset for \((P, \mathcal{X}_k, \tilde{D})\); see Section 1.4.

(iii) \((S, v)\) can be computed in \(O(n \log(n)(k)^m)\) time.

Proof. (i): Let \(J\) denote the number of while iterations in Algorithm 2, and for every \(j \in [J]\) let \(P^j, \mathcal{P}^m_j\) and
$B_m(j)$ denote respectively the sets $P^0$, $P^m$ and $B^m$ at the $j$th while iteration of Algorithm 2.

By Line 7 of Algorithm 1, we observe that the output set $P^m$ is of size $|P^m| \leq \frac{|P|}{(bk)^m}$ for some constant $b$, where $P$ is the input set to the algorithm. Therefore, the size of $P^m_j$ returned at Line 7 of algorithm 2 in the $j$th while iteration is

$$|P^m_j| \geq \frac{|P^0_j|}{(bk)^m}. \quad (48)$$

By (48) and Line 11 of Algorithm 2, we obtain that

$$|P^0_{(j+1)}| \leq |P^0_{(j)}| - |P^m_{(j)}| = \left| P^0_{(j)} \right| \left( 1 - \frac{1}{(bk)^m} \right)^j$$

$$= \left| P^0_{(1)} \right| \left( 1 - \frac{1}{(bk)^m} \right)^j$$

$$= n \left( 1 - \frac{1}{(bk)^m} \right)^j,$$

where the second derivation is by (48). Combining that $|P^0_{(j)}| \geq 1$ with (49) we conclude that

$$J \leq (bk)^m \log n. \quad (50)$$

By Theorem 3.1, a sample $S$ of $|S| \leq \frac{k!}{\varepsilon^d} (\log (t)d' + \log \left( \frac{1}{\delta} \right))$ is an $\varepsilon$-coreset for (the sets clustering query space) $(P, X_k, \hat{D})$. Therefore, by Theorem 3.1, the pair $(S, \nu)$ computed at Lines 17–19 satisfies Property (ii) of Theorem 4.2.

**Computational time.** Consider a call

**RECURSIVE-ROBUST-MEDIAN**($P, k$) to Algorithm 1 where $P$ is an $(n, m)$-set. The $i$th iteration of the for loop at Line 4 takes $O\left(n \left( \frac{1}{44b} \right)^{i-1} + k^4 \right)$ time. Summing over all the $m$ iterations yields a total running time of $O(nm^k k^4)$.

Consider the call $(P^m, B^m) := \mathbf{RECURSIVE-ROBUST-MEDIAN}(P^0, k)$ at Line 7 of Algorithm 2, which dominates the running time of this algorithm. This call is made $J$ times (in each of the $J$ iterations of the while loop). The set $P^0$ at the $i$th call is of size $s_i = O\left(n \left( \frac{1}{44b} \right)^{i-1} \right)$. Therefore, the $i$th such call takes $O(s_i + m^k)$ time. Summing this running time over every $i \in [J]$, where $J \leq (bk)^m \log n$ by (50), yields a total running time of

$$J \cdot m^k + n \sum_{i=1}^{J} \left( 1 - \frac{1}{(bk)^m} \right)^{i-1} \in O\left(n \log(n)(bk)^m \right).$$

**C. Polynomial Time Approximation Scheme**

The following theorem states that given $n$ polynomials in $d$ (constant number of) variables of constant degree, then the space $\mathbb{R}^d$ can be decomposed into a polynomial ($n^d$) number of cells, such that for every $d$ variables $C$ from the cell the sign sequence of all the polynomials is the same cell.

**Theorem C.1** (Theorem 3.4 in (Chazelle et al., 1991)). Let $d$ be a constant and let $\mathcal{F} = \{p_1, \ldots, p_n\}$ be a set of $n$ multivariate polynomials of constant degree with range $\mathbb{R}^d$ and image $\mathbb{R}$. It is possible to split $\mathbb{R}^d$ into $O(n^{2d-2})$ cells $\Delta_i$, with the property that for every polynomials $p_i$, and every cell $\Delta_i$ it holds that $p_i|_{\Delta_i}$ is either positive, negative, or equal to 0 on the entire cell $\Delta_i$. This decomposition, including a set of points $A = \{a_i\}$ with $a_i \in \Delta_i$ can be found in time $O(n^{2d-1} \log n)$.

**C.1. Proof of Theorem 4.4**

**Theorem 4.4.** Let $P$ be an $(n, m)$-set in $\mathbb{R}^d$, $w : P \to [0, \infty)$ be a weights function, $k \geq 1$ be an integer, $\alpha \geq 1$ and $\delta \in [0, 1)$. Let $D$ be a loss function as in Definition 2.1 for $\mathcal{X} = \mathbb{R}^d$. Let ALG be an algorithm that solves the case where $k = m = 1$, i.e., it takes as input a set $Q \subseteq \mathbb{R}^d$.
Sets Clustering

\[X, a \text{ weights function } u : Q \to [0, \infty) \text{ and the failure probability } \delta, \text{ and in time } T(n) \text{ outputs } \hat{c} \in X \text{ that with probability at least } 1 - \delta \text{ satisfies } \sum_{q \in Q} u(q) \cdot \hat{D}(q, \hat{c}) \leq \alpha \cdot \min_{c \in C} \sum_{q \in Q} u(q) \cdot \hat{D}(q, c). \]

By Theorem C.1, we can decompose \( \mathbb{R}^{dk} \) into \( |\Delta(F)| = (nk)^{O(dk)} \) cells \( \{\Delta_j\} \), such that the sign of each polynomial \( p_i \in F \) in an entire cell \( \Delta_j \in \Delta(F) \) is the same, i.e., the sign sequence of all the polynomials in \( F \) is the same over the entire cell \( \Delta' \). Hence, the number of different such sign sequences is at most the number of different cells, which is \( (nk)^{O(dk)} \).

By iterating over every cell \( \Delta' \in \Delta(F) \) and taking the sign sequence of the polynomials in \( F \) in this cell, we would have covered all the different sign sequences, which correspond to all the feasible clustering options of \( P \) into \( k \) clusters. For each option we can evaluate the total cost as described above, and pick the clustering with the smallest total cost.

The running time of such an algorithm is dominated by the computation of such an arrangement of \( \mathbb{R}^{dk} \), and by calling \( ALG |\Delta(F)| \) times; once for each region \( \Delta' \in \Delta(F) \). Computing this arrangement takes \( nmk^{O(dk)} \) time by Theorem C.1 and produces \( |\Delta(F)| = (nk)^{O(dk)} \) cells. Now it takes \( T(n) \cdot (nmk)^{O(dk)} \) total time for the calls to \( ALG \).

C.2. Proof of Corollary 4.5

**Corollary 4.4** (PTAS for sets-\( k \)-means). Let \( \mathcal{P} \) be an \((n, m)\)-set, \( k \geq 1 \) be an integer, and put \( \varepsilon \in (0, \frac{1}{2}] \) and \( \delta \in (0, 1) \). Let OPT be the cost of the sets-\( k \)-means. Then in \( n \log(n)(k)^m + \left(\frac{\log n}{\varepsilon} dnm^k \right)^{O(dk)} \) time we can compute \( \hat{C} \in X_k \) such that with probability at least \( 1 - k \cdot \delta \),

\[
\min_{P \in \mathcal{P}, \hat{c} \in C} \|p - c\|^2 \leq (1 + 4\varepsilon) \cdot \text{OPT}.
\]

**Proof.** We will first compute a coreset for the input \( \mathcal{P} \) and the given cost function \( \hat{D} \) and query set \( X_k \), and then find the sets-\( k \)-means for the (weighted) coreset using Theorem 4.4.

Recall that in this sets-\( k \)-means problem, \( \hat{D}(P, C) = \min_{P \in \mathcal{P}, c \in C} \|p - c\|^2 \) for every \( P, C \subseteq \mathbb{R}^d \).

Let \((S, v)\) be an output of a call to \( \text{CORESET}(\mathcal{P}, k, \varepsilon, \delta) \); see Algorithm 2. Then by Theorem 4.2, \((S, v)\) is an \( \varepsilon \)-coreset for \((\mathcal{P}, X_k, \hat{D})\) of size \( |S| \in O\left(\left(\frac{mdk \log n}{\varepsilon} \right)^2 \right) \) with probability at least \( 1 - \delta \) which is computed in \( O(n \log(n)(bk)^m) \) time; see Section 1.4.

Let \( Q \subseteq X \) be a set of size \(|Q| = n\) and let \( u : Q \to [0, \infty) \) be a weights function. Let \( ALG \) be an algorithm that takes \( Q \) and \( u \) as input and returns the point \( c^* := \frac{\sum_{q \in Q} u(q) \cdot q}{\sum_{q \in Q} u(q)} \in X \).

Observe that \( c^* \) minimizes its sum of weighted squared
distances to the points of \( Q \), i.e.,
\[
\sum_{q \in Q} u(q) \tilde{D}(q, c^*) = \sum_{q \in Q} u(q) \| q - c^* \|^2
\]
\[
= \min_{c \in \mathcal{X}} \sum_{q \in Q} u(q) \| q - c \|^2 = \min_{c \in \mathcal{X}} \sum_{q \in Q} u(q) \tilde{D}(q, c).
\]

Furthermore, observe that \( c^* \) can be computed in \( O(n) \) time.

Plugging \( P = S \), \( w = v = Q \), \( u \), \( \alpha = 1 \) and \( T(|S|) = O(|S|) \) in Theorem 4.4 yields that in \((|S| \log n)^{O(dk)} \) time we can compute \( \hat{C} \in \mathcal{X}_k \) such that with probability at least \( 1 - k \cdot \delta \),
\[
\sum_{P \in S} v(P) \cdot \tilde{D}(P, \hat{C}) = \min_{C \in \mathcal{X}_k} \sum_{P \in S} v(P) \cdot \tilde{D}(P, C). \tag{51}
\]

Hence, the total running time for obtaining \( \hat{C} \) is \((|S| \log n)^{O(dk)} + O(n \log(n)(bk)^m) \).

Corollary 4.5 now holds as
\[
\sum_{P \in \mathcal{P}} \min_{p \in P} \min_{c \in \hat{C}} \| p - c \|^2 = \sum_{P \in \mathcal{P}} \tilde{D}(P, \hat{C})
\]
\[
\leq \frac{1}{1 - \varepsilon} \sum_{P \in \mathcal{S}} v(P) \cdot \tilde{D}(P, \hat{C}) \tag{52}
\]
\[
\leq (1 + 2\varepsilon) \sum_{P \in \mathcal{S}} v(P) \cdot \tilde{D}(P, \hat{C}) \tag{53}
\]
\[
= (1 + 2\varepsilon) \sum_{C \in \mathcal{X}_k} \sum_{P \in \mathcal{S}} v(P) \cdot \tilde{D}(P, C) \tag{54}
\]
\[
\leq (1 + 2\varepsilon)(1 + \varepsilon) \sum_{C \in \mathcal{X}_k} \sum_{P \in \mathcal{P}} \tilde{D}(P, C) \tag{55}
\]
\[
\leq (1 + 4\varepsilon) \sum_{C \in \mathcal{X}_k} \sum_{P \in \mathcal{P}} \tilde{D}(P, C) \tag{56}
\]
\[
= (1 + 4\varepsilon) \sum_{C \in \mathcal{X}_k} \min_{P \in \mathcal{P}} \min_{c \in C} \| p - c \|^2,
\]
where (52) and (55) hold since \((S, v)\) is an \( \varepsilon \)-coreset for \((\mathcal{P}, \mathcal{X}_k, \hat{D})\), (53) and (56) hold since \( \varepsilon \leq \frac{1}{2} \) and (54) is by (51).

\begin{algorithm}[H]
\caption{PTAS(\( P, w, k, \text{ALG} \))}
\begin{algorithmic}[1]
\State \textbf{Input}: An \((n, m)\)-set \( P \), a weights function \( w : \mathcal{P} \to [0, \infty) \), a positive integer \( k \), and an algorithm ALG as in Theorem 4.4.
\State \textbf{Output}: A set \( \hat{C} \in \text{arg min}_{C \in \mathcal{X}_k} \sum_{P \in \mathcal{P}} w(P) \tilde{D}(P, C) \); see Theorem 4.4.
\State Identify \( P = \{ P_1, \cdots, P_n \} \) where \( P_i = \{ p_1^i, \cdots, p_m^i \} \) for every \( i \in [n] \).
\State Define \( w'(p) := w(P) \) for every \( p \in P \) and \( P \in \mathcal{P} \).
\State Let \( x = (x_1^T \cdots x_k^T)^T \in \mathbb{R}^{dk} \) be a vector of \( dk \) unknowns.
\State \textbf{for} every \( i, j \in [n], j, j' \in [m], \ell, \ell' \in [k] \) \textbf{do}
\State \hspace{1em} \( p_{i,j,\ell,j',\ell'}(x) = \| p_j^i - x_\ell \|^2 - \| p_j'^i - x_{\ell'} \|^2 \)
\State \hspace{1em} \text{A polynomial of degree 2 containing up to 2dk unknowns from \( x \). If this polynomial is positive if \( p_{j,\ell} \)
\State \hspace{1em} \text{is closer to} \( x_\ell \) \text{than the distance between} \( p_j^i \text{ and} \ x_\ell \).
\State \textbf{end for}
\State Compute a decomposition of \( \mathbb{R}^{dk} \) into cells \( \Delta(\mathcal{F}) = \{ \Delta_1 \} \) as described in Theorem C.1, and let \( A \) contain a representative \( a \in \Delta_\ell \) from each cell \( \Delta_\ell \in \Delta(\mathcal{F}) \).
\State \textbf{min} \( = \infty \)
\State \textbf{for} every \( a \in A \) \textbf{do}
\State \hspace{1em} \( \text{sum} = 0 \)
\State \hspace{1em} \textbf{for} every \( \ell \in [k] \) \textbf{do}
\State \hspace{2em} \( \mathcal{C}_\ell := \left\{ p_j^i \bigg| \begin{array}{l}
i \in [n], j \in [m] \text{s.t.} \\
\forall j' \in [m], \ell' \in [k] \\
\text{sign} (p_{i,j,\ell,j',\ell'}(a)) = -1
\end{array} \right\} \)
\State \hspace{2em} \{ The points of cluster number \( \ell \) defined by the sign sequence of the cell representative \( a \in A \). \}
\State \hspace{2em} \( \hat{c}_\ell, \text{cost}_\ell := \text{ALG}(\mathcal{C}_\ell, w') \).
\State \hspace{2em} Compute the optimal center \( (k = 1) \) \( \hat{c}_\ell \) of the set \( \mathcal{C}_\ell \subseteq \mathbb{R}^d \) \((m = 1) \) and its cost \( \text{cost}_\ell \), for a given cost function.\}
\State \hspace{2em} \text{sum} = \text{sum} + \text{cost}_\ell
\State \hspace{1em} \textbf{end for}
\State \hspace{1em} \textbf{if} \( \text{sum} < \text{min} \) \textbf{then}
\State \hspace{2em} \text{min} = \text{sum}
\State \hspace{2em} \text{C} = \{ \hat{c}_1, \cdots, \hat{c}_k \}
\State \hspace{1em} \textbf{end if}
\State \hspace{1em} \textbf{end for}
\State Return \( \hat{C} \)
\end{algorithmic}
\end{algorithm}

C.3. Suggested implementation

In this section we give a suggested implementation for the constructive proof of Theorem 4.4; see Algorithm 4.

Overview of Algorithm 4. Algorithm 4 gets as input a set \( P \) of \( m \)-sets, an integer \( k \geq 1 \), an error parameter \( \varepsilon \in (0, 1) \) and the probability of failure \( \delta \in (0, 1) \). The algorithm returns as output a set \( \hat{C} \in \mathcal{X}_k \) of \( k \) centers that approximate the optimal cost of the \( k \)-means for set

D. Robust Median

D.1. Proof of Lemma 5.1

Algorithm 3 overview: The algorithm relies on the 2 following observations: (i) To compute a robust approximation of the entire data, it suffices to compute a robust approximation of a randomly sampled subset of this data of sufficient size; see Line 4 of Algorithm 3 and Lemma D.1. (ii) If \( b \) is a
robust approximation of some input set of elements, then by
the (weak) triangle inequality for singletons, one of those
elements is a constant factor approximation for \( b \); see Line 5
of Algorithm 3.

**Lemma D.1.** Let \( \mathcal{P} \) be an \((n, m)\)-set, \( k \geq 1, \delta, \gamma \in (0, 1), \)
and \( \tau \in (0, 1/10) \). Pick uniformly, i.i.d. a (multi-)set \( S \) of
\[
|S| = \frac{c}{\tau^2} \left( md^2 + \log \left( \frac{1}{\delta} \right) \right)
\]
elements from \( \mathcal{P} \), where \( c \) is a sufficiently large universal
constant. Then with probability at least \( 1 - \delta \), any \((1 -
\tau)\gamma, \tau, 2)\)-median of \( S \) is also a \((\gamma, 4\tau, 2)\)-median of \( \mathcal{P} \).

**Proof.** For every \( P \in \mathcal{P} \) and \( b \in X_1 \) define \( f_P(b) =
\hat{D}(P, b) \). Let \( F = \{ f_P|P \in \mathcal{P} \} \) and \( F_S = \{ f_P|P \in S \} \).
Observe that by Definition 4.2 in (Feldman & Langberg,
2011), the dimension of the function space \((F, X_1)\) is
equivalent to the dimension \( d' = md^2 \) of the query
space \((\mathcal{P}, X_1, \hat{D})\). Since \( F_S \) is a random sample of
\[
\frac{1}{\tau^2} \left( d' + \log \left( \frac{1}{\delta} \right) \right) = \frac{1}{\tau^2} \left( md^2 + \log \left( \frac{1}{\delta} \right) \right)
\]
sampled i.i.d from \( F \), Lemma D.1 now holds by Theo-
rem 9.6 in (Feldman & Langberg, 2011) which states that
a \((1 - \tau)\gamma, \tau, 2)\)-median of \( F_S \) (which in our case is a
\((1 - \tau)\gamma, \tau, 2)\)-median of \( S \)) is also a \((\gamma, 4\tau, 2)\)-median of \( F \)
(which in our case is a \((\gamma, 4\tau, 2)\)-median of \( \mathcal{P} \)).

**Lemma 5.1** (based on Lemma 9.6 in (Feldman & Langberg,
2011)). Let \( \mathcal{P} \) be an \((n, m)\)-set, \( k \geq 1, \delta, \gamma \in (0, 1) \)
and \((X, \hat{D})\) be as in Definition 2.1. Let \( q \in X \) be the output
of a call to \textsc{median}(\( \mathcal{P}, k, \delta \)); see Algorithm 3. Then with
probability at least \( 1 - \delta \), \( q \) is a \((1/(2k), 1/6, 2)\)-median
for \( P \); see Definition 2.4. Furthermore, \( q \) can be computed
in \( O \left( tb^2k^4 \log^2 \left( \frac{1}{\delta} \right) \right) \) time, where \( t \) is the time it takes to compute \( \hat{D}(P, Q) \) for \( P \in \mathcal{P} \).

**Proof.** Let \( \gamma = 1/(2k) \) and \( \tau = 1/24 \). For a sufficient con-
stant \( b \), the random sample \( S \) in Line 4 satisfies Lemma D.1.
Therefore,
\[
a \left( \frac{23}{(48k)}, 1/24, 2 \right)\text{-median of } S \text{ is also a } \left( \frac{1}{(2k)}, 1/6, 2 \right)\text{-median of } \mathcal{P}.
\]
Let \( q_S^* \) be the \( \left( \frac{23}{(48k)}, 0, 0 \right)\)-median of \( S \), and let \( q_S \)
be the closest point in \( S \) to \( q_S^* \), i.e.,
\[
q_S \in \arg \min_{q \in Q; Q \subseteq S} \hat{D}(q_S^*, q).
\]
By the weak triangle inequality from Lemma 2.2, we have
\( \hat{D}(P, q_S^*) \leq 2p \hat{D}(P, q_S) \) for every \( P \in S \),
i.e., that \( q_S^* \) is a 2-approximation for \( q_S \). This yields
that \( q_S^* \) is a \( \left( \frac{23}{(48k)}, 0, 2 \right)\)-median of \( S \), which is also
a \( \left( \frac{23}{(48k)}, 1/6, 2 \right)\)-median of \( S \). Hence, one of the points
of \( S \) is a \( \left( \frac{23}{(48k)}, 1/6, 2 \right)\)-median of \( S \). Therefore, the

**Figure 6. exact-mean via sets Voronoi diagram.** A set of
\( n = 4 \) pairs on the plane \((m = d = 2)\) and its sets Voronoi
diagram which is computed as follows: (i) A set voronoi diagram
is computed for each pair \((m\text{-set})\) to obtain a set of hyperplanes,
(ii) an arrangement of those hyperplanes is then computed, which
results in a partition of \( \mathbb{R}^2 \) into the cells which are presented above.
Each cell corresponds to a selection of representatives, one from
each pair. The sets-mean \( c^* \) (solid star) is also the 1-mean of the
representative points shown in solid circles, which correspond to
this Voronoi cell. Any other point (empty star) inside the same
Voronoi cell as \( c^* \) admits the same set of representatives.
Therefore, to compute \( c^* \), it suffices to exhaustive search over all the
Voronoi.

The computation time of Algorithm 3 is dominated by
Line 5, which can be implemented in \( t|S|^2 =
tb^2k^4 \log^2 \left( \frac{1}{\delta} \right) \) time by simply computing the pairwise distances
between every two sets in \( S \) and using order statistics.

**E. Implemented Algorithms**

**exact-mean***(\( \mathcal{P} \))** is implemented by what we call sets
Voronoi diagram; see Fig. 6.

**k-means***(\( \mathcal{P}, k \)). We focused on the sets-\( k \)-means case
(see Section 1 and Table 1), where the clustering algo-
rum we applied is a modified version of the well
know Lloyd algorithm (Lloyd, 1982) as follows. The
algorithm starts by an initial \( k \) random centers \( C \subseteq
\{ p \in P | P \in \mathcal{P} \} \). It then assigns every \( P \in \mathcal{P} \) to its closest
center \( c_P = \arg \min_{c \in C} \hat{D}(P, c) \). Finally, it replaces every
\( c \in C \) with the sets-mean of the (possibly weighted) sets
\( \{ P \in \mathcal{P} | c_P = c \} \) in its cluster. It repeats this process till
convergence, but no more than 12 iterations. The sets-mean
is computed as follows.

**approx-mean***(\( \mathcal{P}, t \)). As explained in Section 1, computing
the sets-mean \( c^* \) is a non-trivial and time consuming
task. However, at least \(|\mathcal{P}|/2\) of the input sets \( P \in \mathcal{P} \) satisfy
that \( \hat{D}(P, c^*) \leq 2\sum_{Q \subseteq \mathcal{P}\backslash\{P\}} \hat{D}(Q, c^*) \). By the triangle inequality
for singletons (Lemma 2.2), it follows immediately that the
closest point \( p \in P \) to \( c^* \) is a 3-approximation for \( c^* \). Therefore,
with probability at least 1/2, one of the points of a
randomly sampled input set is a good approximation. We can amplify this probability by sampling \( t \geq 1 \) such sets.

**Handling sets of different sizes.** For example in dataset (ii), each newspaper \( P_i \) consists of different number of paragraphs and hence is represented by a different number \( |P_i| \) of vectors. Let \( z \) denote the maximal such set size. To compute a coreset for such dataset \( \mathcal{P} \), we first partition \( \mathcal{P} \) into \( z \) sets \( \mathcal{P} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_z \) where \( \mathcal{P}_i \) contains all the sets \( P \in \mathcal{P} \) of size \( |P| = i \). Then, for every \( i \in \{z\} \), we plug \( \mathcal{P}_i = \mathcal{P}_i \) at Lines 5–14 of Algorithm 2 to compute \( s(P) \) for every \( P \in \mathcal{P}_i \). In other words, we compute the sensitivity bound for each set on its own. We compute the total sensitivity \( t_i := \sum_{P \in \mathcal{P}} s_i(P) \) of each set \( \mathcal{P}_i \) and \( t := \sum_{i \in [m]} t_i \) to be their total. We then simply perform Lines 17–21.