1 Coreset for Optimal Decision Tree

1.1 Notation

Let ∞ be a value s.t. $-\infty < r < \infty$ for every real $r \in \mathbb{R}$ and let d and H be a pair of positive integers.

The document uses numpy-like notation: a vector is a matrix with one column; a matrix is indexed using square brackets; matrix indexes start from 0; [] is a matrix with 0 rows and 0 columns; for a pair of matrices A and B: $[A; B]$ is their concatenation; for an integer i and a matrix A: $A[:, i]$ is the i-th column of A and $A[i, :]$ is the i-th row of A (which are also matrices).

$\text{sort}(A, i)$ is the matrix A with its rows reordered to make the values of i-th column grow ascendingly and $\text{reverse}(A)$ is the matrix A with its rows reversed (the first row becomes the last); a matrix whose element's values are either True or False is called *boolean*; binary operations on a matrix and a scalar or a set are performed piecewise on the elements of the matrix; for a boolean array B and a matrix A with the same number of rows, the matrix $A[B, :]$ is the matrix A with rows whose index not in $\{i \mid B[i] = \text{True}\}$ removed; a set of n points $S \subset \mathbb{R}^d$ and an $n \times d$ real matrix M are *equivalent* if for every $p \in P$ there is $i \in [n-1] \cup \{0\}$ such that j-th coordinate of p equals the value of $M[i, j-1]$ for every integer $j \in [d]$; for every matrix M, $\text{set}(M)$ is the equivalent set of M and for every set S, $\text{matrix}(S)$ is an arbitrarily chosen equivalent matrix of S.

Let $\mathcal{A} : [H] \cup \{0\} \to [d-1] \cup \{0\}$ be a function which assigns split axis to each tree height.

1.2 Definitions

For every positive integer n, for every $n \times d$ real matrix P, for every positive integer h and for every real t we define:

$$
\begin{aligned}
opt(P, 0) &:= \max_{p,q \in \text{set}(P)} \|p - q\| \\
\mathcal{L}(P, h, t) &:= P[P[:, \mathcal{A}(h)] \le t, :] \\
\mathcal{R}(P, h, t) &:= P[P[:, \mathcal{A}(h)] > t, :] \\
opt(P, h) &:= \max_{t \in \mathbb{R}} \sum_{Q \in \{\mathcal{L}(P,h,t), \mathcal{R}(P,h,t)\}} opt(Q, h-1)
\end{aligned}
$$

and

$$lcost(P,h,t) := opt(\mathcal{L}(P,h,t), h-1)$$
$$rcost(P,h,t) := opt(\mathcal{R}(P,h,t), h-1)$$
$$cost(P,h,t) := lcost(P,h,t) + rcost(P,h,t)$$

1.3 Algorithm

Algorithm 1.1: $\textsc{Base}(P)$

$PQ \leftarrow \left\{(p,q) \in \text{set}(P)^2 \mid \|p-q\| = cost(P)\right\}$
$P \leftarrow \text{matrix}(\{p \mid (p,q) \in PQ\})[0,:]$
$Q \leftarrow \text{matrix}(\{q \mid (p,q) \in PQ\})[0,:]$
return $([P;Q])$

Algorithm 1.2: $\text{OT-}\textsc{Coreset}(P, \mathcal{A}, \varepsilon, h)$

if $h = 0$
  then return $(\textsc{Base})(P)$
  else
    $C \leftarrow [], Q \leftarrow [], L \leftarrow [], X \leftarrow \{P\}$
    for each $M \in \{\text{sort}(P), \text{reverse}(\text{sort}(P))\}$
      do for $i \leftarrow 0$ **to** $|\text{set}(M)| - 1$
        do
          $p \leftarrow M[i,:]$
          if $opt([Q;p], h-1) \geq (1+\varepsilon) \cdot opt(L, h-1)$
            then
              output $(p[\mathcal{A}(h)])$
              $L \leftarrow Q$
              $R \leftarrow \text{matrix}(\text{set}(P) \setminus \text{set}([L;p]))$
              $X \leftarrow X \cup \{L, R, [L;p], [R;p]\}$
          $Q \leftarrow [Q;p]$
    for each $x \in X$
      do
        $c \leftarrow \text{OT-}\textsc{Coreset}(x, \mathcal{A}, \varepsilon, h-1)$
        $C \leftarrow [C;c]$
return (C)

1.4 Analysis

Theorem 1. *Let P be an $n \times d$ real matrix, let $h \leq H$ be a nonnegative integer and let t be a real. Then for $C = \text{OT-}\textsc{Coreset}(P, \mathcal{A}, \varepsilon, h)$, $cost(C,h,t)$ is an $(1+\varepsilon)^h$-approximation of $cost(P,h,t)$.*

Proof. We claim that if A is a subset of P then for every nonnegative integer $h \leq H$, for every real t and for $C = \text{OT-}\textsc{Coreset}(P, \mathcal{A}, \varepsilon, h)$:

$$(1-\varepsilon)^h cost(P, h, t) \leq cost(C \cup A, h, t) \leq (1+\varepsilon)^h cost(P, h, t)$$

and we prove our claim by induction on h.

For $h = 0$, let m be the number of rows printed by the algorithm and let $t_1, \ldots, t_m \in \mathbb{R}$ be the output of the algorithm sorted ascendingly; let $t_0 = -\infty$ and $t_{m+1} = \infty$ and let $T = \{t_0, t_1, \ldots, t_{m+1}\}$.

Let $C_\ell = \text{set}\left(\textsc{Base}(\mathcal{L}(P, 0, t))\right)$ and let $C_R = \text{set}\left(\textsc{Base}(\mathcal{R}(P, 0, t))\right)$.

If $t \in T$ then by algorithm construction $C_\ell, C_r \subseteq C$ and therefore

$$\begin{aligned} lcost(C, 0, t) &= lcost(P, 0, t) \\ rcost(C, 0, t) &= rcost(P, 0, t) \end{aligned}$$

and therefore $lcost(C \cup A, 0, t) = lcost(P, 0, t)$ and $rcost(C \cup A, 0, t) = rcost(P, 0, t)$ since $C \subseteq C \cup A \subseteq P$ and since $lcost, rcost$ are monotonic. Therefore

$$\begin{aligned} cost(C \cup A, 0, t) &= lcost(C \cup A, 0, t) + rcost(C \cup A, 0, t) \\ &= lcost(P, 0, t) + rcost(P, 0, t) = cost(P, 0, t). \end{aligned}$$

Otherwise, $t \notin T$. Let $i \in [m+1] \cup \{0\}$ be the largest index such that $t_i \leq t$ and let $t_\ell = t_i$ and $t_r = t_{i+1}$.

By algorithm construction we have

$$lcost(P, 0, t_\ell) \leq lcost(P, 0, t) \leq lcost(P, 0, t_r) \leq (1+\varepsilon) lcost(P, 0, t_\ell)$$

and

$$rcost(P, 0, t_r) \leq rcost(P, 0, t) \leq rcost(P, 0, t_\ell) \leq (1+\varepsilon) rcost(P, 0, t_r).$$

Therefore, since $t_\ell, t_r \in T$

$$\begin{aligned} lcost(C \cup A, 0, t) &\leq lcost(C \cup A, 0, t_r) = lcost(P, 0, t_r) \\ &\leq (1+\varepsilon) cost(P, 0, t_\ell) \leq (1+\varepsilon) cost(P, 0, t) \end{aligned}$$

and

$$\begin{aligned} rcost(C \cup A, 0, t) &\leq rcost(C \cup A, 0, t_\ell) = rcost(P, 0, t_\ell) \\ &\leq (1+\varepsilon) rcost(P, 0, t_r) \leq (1+\varepsilon) rcost(P, 0, t) \end{aligned}$$

and therefore

$$cost(C \cup A, 0, t) \leq (1+\varepsilon) cost(P, 0, t).$$

By algorithm construction $lcost(P, 0, t_r) \leq lcost(P, 0, t_\ell) + \varepsilon lcost(P, 0, t_\ell)$ and therefore

$$lcost(P, 0, t_\ell) \geq lcost(P, 0, t_r) - \varepsilon lcost(P, 0, t_\ell) \geq (1 - \varepsilon) lcost(P, 0, t_r)$$

and similary $rcost(P, 0, t_r) \geq (1 - \varepsilon) rcost(P, 0, t_\ell)$. Therefore

$$\begin{aligned} lcost(C \cup A, 0, t) &\geq lcost(C \cup A, 0, t_\ell) = lcost(P, 0, t_\ell) \\ &\geq (1 - \varepsilon) lcost(P, 0, t_r) \geq (1 - \varepsilon) lcost(P, 0, t). \end{aligned}$$

and

$$\begin{aligned} rcost(C \cup A, 0, t) &\geq rcost(C \cup A, 0, t_r) = rcost(P, 0, t_r) \\ &\geq (1 - \varepsilon) rcost(P, 0, t_\ell) \geq (1 - \varepsilon) rcost(P, 0, t) \end{aligned}$$

and therefore

$$cost(C \cup A, 0, t) \geq (1 - \varepsilon) cost(P, 0, t).$$

We now assume that the claim is correct for $0 \leq h < H$, that is that for $C = \text{OT-Coreset}(P, \mathcal{A}, \varepsilon, h)$ and for every $A \subseteq P$ the following holds

$$(1 - \varepsilon)^h cost(P, h, t) \leq cost(C \cup A, h, t) \leq (1 + \varepsilon)^h cost(P, h, t)$$

and prove that it is also holds for $h+1$, when $C = \text{OT-Coreset}(P, \mathcal{A}, \varepsilon, h+1)$.

Note, that since the claim holds for every t we can also say that

$$(1 - \varepsilon)^h opt(P, h) \leq opt(C \cup A, h) \leq (1 + \varepsilon)^h opt(P, h).$$

Let m be the number of rows printed by the algorithm and let $t_1, \ldots, t_m \in \mathbb{R}$ be the printed output of the algorithm; let $t_0 = -\infty$ and $t_{m+1} = \infty$ and let $T = \{t_0, t_1, \ldots, t_{m+1}\}$.

Let

$$\begin{aligned} C_\ell &= \text{set}\left(\text{OT-Coreset}(\mathcal{L}(P, 0, t), t), \mathcal{A}, \varepsilon, h\right) \\ C_R &= \text{set}\left(\text{OT-Coreset}(\mathcal{R}(P, 0, t), \mathcal{A}, \varepsilon, h)\right). \end{aligned}$$

If $t \in T$ then by algorithm construction $C_\ell, C_r \subseteq C$. Let $A_\ell = C \setminus C_\ell$ and let $A_r = C \setminus C_r$.

By definition, $lcost(P, h+1, t) = opt(\mathcal{L}(P, 0, t), h)$ and $opt(\mathcal{L}(P, 0, t), h+1) = opt(\mathcal{L}(C_\ell \cup A_\ell, 0, t), h+1)$. Therefore, by induction assumption

$$\begin{aligned} opt(\mathcal{L}(P, 0, t), h)(1 - \varepsilon)^h &\leq opt(\mathcal{L}(C_\ell \cup A_\ell, 0, t), h) \\ &\leq (1 + \varepsilon)^h opt(\mathcal{L}(P, 0, t), h) \end{aligned}$$

and similarly,

$$\begin{aligned} opt(\mathcal{R}(P,0,t),h-1)(1-\varepsilon)^h &\leq opt(\mathcal{R}(C_r \cup A_r,0,t),h) \\ &\leq (1+\varepsilon)^h opt(\mathcal{R}(P,0,t),h) \end{aligned}$$

and therefore

$$\begin{aligned} cost(P,h+1,t)(1-\varepsilon)^h &\leq cost(P,h+1,t) \\ &\leq (1+\varepsilon)^h cost(P,h+1,t). \end{aligned}$$

If $t \notin T$ let $i \in [m+1] \cup \{0\}$ be the largest index such that $t_i \leq t$ and let $t_\ell = t_i$ and $t_r = t_{i+1}$.

By algorithm construction we have

$$lcost(P,0,t_\ell) \leq lcost(P,0,t) \leq lcost(P,0,t_r) \leq (1+\varepsilon) lcost(P,0,t_\ell)$$

and

$$rcost(P,0,t_r) \leq rcost(P,0,t) \leq rcost(P,0,t_\ell) \leq (1+\varepsilon) rcost(P,0,t_r).$$

Therefore, since $t_\ell, t_r \in T$

$$\begin{aligned} lcost(C \cup A,0,t) &\leq lcost(C \cup A,0,t_r) \leq (1+\varepsilon)^h lcost(P,0,t_r) \\ &\leq (1+\varepsilon)^h (1+\varepsilon) cost(P,0,t_\ell) \leq (1+\varepsilon)^{h+1} cost(P,0,t) \end{aligned}$$

and

$$\begin{aligned} rcost(C \cup A,0,t) &\leq rcost(C \cup A,0,t_\ell) \leq (1+\varepsilon)^h rcost(P,0,t_\ell) \\ &\leq (1+\varepsilon)^h (1+\varepsilon) rcost(P,0,t_r) \leq (1+\varepsilon)^{h+1} rcost(P,0,t) \end{aligned}$$

and therefore

$$cost(C \cup A,0,t) \leq (1+\varepsilon)^{h+1} cost(P,0,t).$$

By algorithm construction $lcost(P,0,t_r) \leq lcost(P,0,t_\ell) + \varepsilon lcost(P,0,t_\ell)$ and therefore

$$lcost(P,0,t_\ell) \geq lcost(P,0,t_r) - \varepsilon lcost(P,0,t_\ell) \geq (1-\varepsilon) lcost(P,0,t_r)$$

and similarly $rcost(P,0,t_r) \geq (1-\varepsilon) rcost(P,0,t_\ell)$. Therefore

$$\begin{aligned} lcost(C \cup A,0,t) &\geq lcost(C \cup A,0,t_\ell) \geq (1-\varepsilon)^h lcost(P,0,t_\ell) \\ &\geq (1-\varepsilon)^h (1-\varepsilon) lcost(P,0,t_r) \geq (1-\varepsilon)^{h+1} lcost(P,0,t). \end{aligned}$$

and

$$\begin{aligned} rcost(C \cup A, 0, t) &\geq rcost(C \cup A, 0, t_r) \geq (1-\varepsilon)^h rcost(P, 0, t_r) \\ &\geq (1-\varepsilon)^h (1-\varepsilon) rcost(P, 0, t_\ell) \geq (1-\varepsilon)^{h+1} rcost(P, 0, t) \end{aligned}$$

and therefore

$$cost(C \cup A, 0, t) \geq (1-\varepsilon) cost(P, 0, t).$$

$\square$

2 Approximate Decision Tree

In this section the value of an empty sum is 0. Let $\varepsilon \in (0,1) \subset \mathbb{R}$ be a real number in the open interval $(0,1)$. Let $d, n \in \mathbb{Z}_{>0}$ be a pair of positive integers. For every $i \in [d]$ let $g_i \subset \mathbb{R}$ be a finite set of a coordinates and let $G = g_1 \times g_2 \times \cdots \times g_d$.

G imposes a grid, for instance we may choose a d positive real numbers $\sigma_1, \ldots, \sigma_d \in \mathbb{R}_{>0}$ and d positive integers $m_1, \ldots, m_d \in \mathbb{Z}_{>0}$ and define

$$g_i := \left\{\sigma_i, (1+\varepsilon)\sigma_i, (1+\varepsilon)^2\sigma_i, \ldots, (1+\varepsilon)^{m_i}\sigma_i\right\}$$

to make the grid exponentially increasing.

For $i \in [d]$ and $p \in \mathbb{R}^d$ we denote the i-th coordinate of p by $p[i]$. We may treat a point $p \in \mathbb{R}^d$ as a d-tuple by writing $p = (p[1], p[2], \ldots, p[d])$. For a pair of points $p, q \in \mathbb{R}^d$ we define

$$\gamma(q_1, q_2) := \left\{ p \in \mathbb{R}^d \mid \forall i \in [d] : q_1[i] < p[i] \leq q_2[i] \right\}.$$

Let

$$\begin{aligned} \mu(P, i) &:= \frac{1}{|P|} \cdot \sum_{p \in P} p[i] \\ \lambda(P, i) &:= \min\{p[i] \mid p \in P\} \\ \rho(P, i) &:= \max\{p[i] \mid p \in P\} \end{aligned}$$

For $i \in [d]$ let

$$A = \gamma(\lambda(G, 1), \ldots, \lambda(G, d)), (\rho(G, 1), \ldots, \rho(G, d)).$$

For a compact of points $P \subset A$ and an integer $i \in [d]$ we define

$$\alpha(P) := (\mu(P,1), \mu(P,2), \dots, \mu(P,d))$$
$$\beta(P) := \sum_{p \in P} \|p - \alpha(P)\|^2$$

For a point $p \in A$ we define

$$\Gamma = \left\{(q_1, q_2) \in G^2 \mid \forall i \in [d] : q_1[i] < q_2[i]\right\}$$
$$\Phi := \{\gamma(q_1, q_2) \mid q_1, q_2 \in \Gamma\}$$
$$\phi(p) := \{C \in \Phi \mid p \in C\}$$

For a set of points $P \subseteq A$ an integer $i \in [d]$ and a real $t \in \mathbb{R}$ we define

$$\mathcal{L}(P, i, t) := \{p \in P \mid p[i] < t\}$$
$$\mathcal{R}(P, i, t) := P \setminus \mathcal{L}(P, i, t)$$

For a positive integer h, *h-tree* is a tuple (t, i, L, R) where $i \in [d]$ is an integer, $t \in g_i$ is a coordinate and L, R are ℓ-tree and r-tree respectively for integers $0 \leq \ell, r < h$ such that either $\ell = h-1$ or $r = h-1$; 0-tree is an empty tuple. For h-tree (t, i, L, R) and a finite non-empty set of points $P \subset A$ we define

$$\operatorname{cost}(P, ()) := \beta(P)$$
$$\operatorname{cost}(P, (t, i, L, R)) := \operatorname{cost}(\mathcal{L}(P, i, t), L) + \operatorname{cost}(\mathcal{R}(P, i, t), R).$$

Finally, we define the function $s : 2^A \times A \to \mathbb{R}$ to be

$$s(P, p) := \sum_{C \in \phi(p)} \frac{\|\alpha(C \cap P) - p\|^2}{\operatorname{cost}(C \cap P, ())}.$$

We will use the following algorithm in the next theorem.

Algorithm 2.1: $\textsc{Leaf}(p, T)$

comment: $p \in A$ and T is an h-tree

$$
\begin{array}{l}
\ell \leftarrow (\lambda(G,1), \ldots, \lambda(G,d)) \\
r \leftarrow (\rho(G,1), \ldots, \rho(G,d)) \\
\textbf{while } T \neq () \\
\quad \textbf{do } \begin{cases} (t, i, L, R) \leftarrow T \\ \textbf{if } p[i] \leq t \\ \quad \textbf{then } \begin{cases} r[i] \leftarrow t \\ T \leftarrow L \end{cases} \\ \quad \textbf{else } \begin{cases} \ell[i] \leftarrow t \\ T \leftarrow R \end{cases} \end{cases} \\
\textbf{return } (\gamma(\ell, r))
\end{array}
$$

Theorem 2. *For every positive integer h, for every set of points $P \subseteq A$, for every h-tree T and for every $p \in P$:*

$$\frac{\|\alpha(\textsc{Leaf}(p,T) \cap P) - p\|^2}{\text{cost}(P,T)} \leq s(P,p)$$

Proof. Let r, ℓ be the variables from Algorithm $\textsc{Leaf}$. These variables are points such that $r, \ell \in G$: the are initialized to be points in G and their coordinates are altered only to values from G.

Hence:

$$\textsc{Leaf}(p,T) = \gamma(\ell, r) \in \Phi$$

On the other hand $p \in \textsc{Leaf}(p,T)$ since from the construction of Algorithm $\textsc{Leaf}$ follows that for every $i \in [d] : \ell[i] \leq p[i] \leq r[i]$.

Let $C = \textsc{Leaf}(p,T)$. Since $C \in \Phi$ and $p \in C$: $C \in \phi(p)$.

Therefore, from the definition of $s(P,p)$ follows that

$$\frac{\|\alpha(C \cap P) - p\|^2}{\text{cost}(C \cap P, ())} \leq s(P,p)$$

On the other hand, from the definition of cost follows that

$$\text{cost}(C \cap P, ()) \leq \text{cost}(P,T)$$

and therefore

$$\frac{\|\alpha(C \cap P) - p\|^2}{\text{cost}(P,T)} \leq \frac{\|\alpha(C \cap P) - p\|^2}{\text{cost}(C \cap P, ())}.$$

Hence

$$\frac{\|\alpha(C \cap P) - p\|^2}{\text{cost}(P, T)} \le s(P, p).$$

$\square$

Theorem 3. *For any finite set of points $P \subset A$:*

$$\sum_{p \in P} s(P, p) = O(m^{2d})$$

Proof. G is a Cartesian product of d sets of cardinality m and therefore $|G| = m^d$. Therefore

$$|\Phi| \le \left|G^2\right| = \binom{m^d}{2} = O(m^{2d})$$

and therefore

$$\sum_{p \in P} s(P, p) = \sum_{p \in P} \sum_{C \in \phi(p)} \frac{\|\alpha(C \cap P) - p\|^2}{\text{cost}(C \cap P, ())} = \sum_{C \in \Phi} \sum_{p \in P \cap C} \frac{\|\alpha(C \cap P) - p\|^2}{\text{cost}(C \cap P, ())}$$
$$= \sum_{C \in \Phi} \frac{\text{cost}(C \cap P, ())}{\text{cost}(C \cap P, ())} \sum_{C \in \Phi} 1 = |\Phi| = O(m^{2d}).$$

$\square$

Theorem 4 (Link to sensitivity article)**.** *Let h be a positive integer. Let $\mathcal{T}$ be a set of all h-trees. For every set of points $P \subseteq A$, for every h-tree T and for every $p \in P$*

$$\sum_{p \in P} s(P, p) = O(m^{2d})$$

and

$$\frac{\|\alpha(\textsc{Leaf}(p, T) \cap P) - p\|^2}{\text{cost}(P, T)} \le s(P, p)$$

and therefore for every positive integer h, for every set of points $P \subseteq A$, for every h-tree T there is a set of weighted points C such that and a function $\text{cost}' : 2^{\mathbb{R} \times \mathbb{R}^d} \times \mathcal{T} \to$ *real such that*

$$(1 - \varepsilon)\text{cost}(P, T) \le \text{cost}'(C, T) \le (1 + \varepsilon)\text{cost}(P, T).$$

3 (No) Coreset For Arbitrary Trees

For positive integer h, *h-tree* is a tuple (t, i, L, R) when t is a real, $i \in [d]$ is an integer and L, R are ℓ-tree and r-tree respectively for integers $0 \le \ell, r < h$

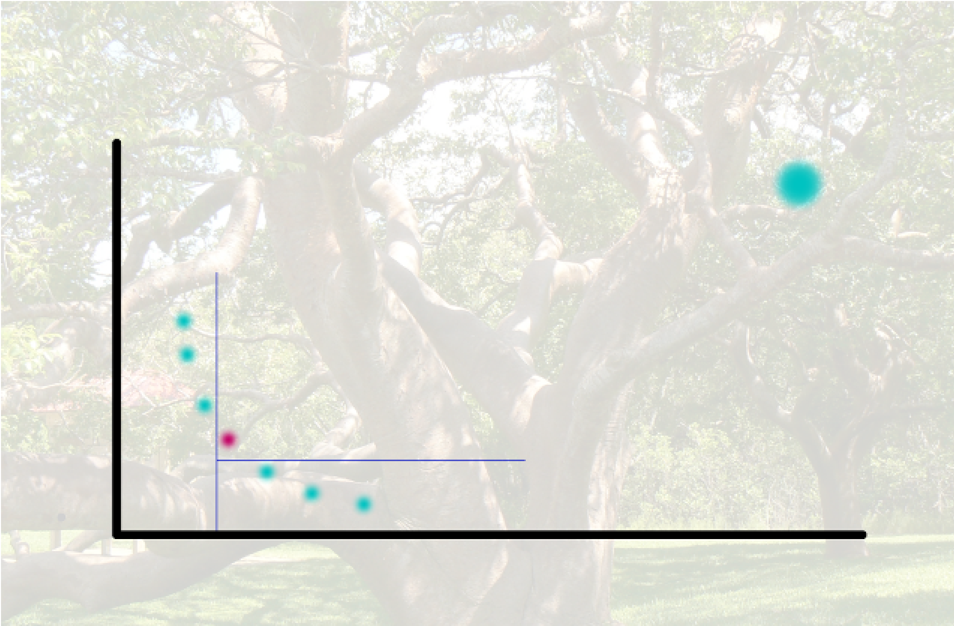

Figure 1: green and red points are in P, green points are also in C, blue lines are tree partitions

such that either $\ell = h - 1$ or $r = h - 1$; 0-tree is an empty tuple. For h-tree (t, i, L, R) and a set of points P:

$$\mathrm{cost}(P, ()) = MSE(P)$$
$$\mathrm{cost}(P, (t, i, L, R)) = \mathrm{cost}(\mathcal{L}(P, i, t), L) + \mathrm{cost}(\mathcal{R}(P, i, t), R)$$

Theorem 5. *Let h be a positive integer larger than 2. There is a set of points P such that for every subset $C \subset P$ there is h-tree T such that*

$$\mathrm{cost}(C, T) \not\geq (1 - \varepsilon)\mathrm{cost}(P, T)$$

Proof. Let $n > 10$ be a positive integer. Let Q be a set of n random points on a left bottom quarter of a circle centered at $(1, 1)$ with radius 1. Let $P = Q \cup \left\{(n \cdot 2^{11}, n \cdot 2^{11})\right\}$. Let $C \subset Q$. There is $p \in P$ such that $p \notin C$. If $p = (n \cdot 2^{11}, n \cdot 2^{11})$ then for $T = (\infty, 0, (\infty, 0, (\infty, 0, (), ()), ()), ())$: $\mathrm{cost}(C, T) \leq n$ and $\mathrm{cost}(P, T) > n \cdot 2^{10}$ and hence

$$\mathrm{cost}(C, T) \not> (1 - \varepsilon)\mathrm{cost}(P, T).$$

Therefore, $p \neq (n \cdot 2^{11}, n \cdot 2^{11})$. Hence, p is one of the points on the quarter circle. Every such point can be separated by an h-tree, see Figure 1. $\square$

4 Coreset For H=1

4.1 Notation

In this section, we use the convention that the value of an empty sum of numbers is 0. For a concatenation of any two row vectors $\phi = (\phi_1, \dots, \phi_m)$ and $\chi = (\chi_1, \dots, \chi_m)$ we denote $(\phi, \chi) = (\phi_1, \dots, \phi_m, \chi_1, \dots, \chi_m)$. For example, for a real number t and a vector $p \in \mathbb{R}^d$, the pair (t, p) is a point in $\mathbb{R}^{d+1}$.

Here an thereafter, we assume that d and k are a pair of positive integers and that $j \leq d$ is a non-negative integer.

Let S be an affine subspace of $\mathbb{R}^d$. For a real number $t \in \mathbb{R}$ we define $S(t)$ to be

$$S(t) := \begin{cases} \{(t,q) \in S\} & |S| > 1 \\ S & \text{otherwise} \end{cases}$$

and we define the *(regression) distance* of a point $(t,p) \in \mathbb{R}^d$ to S to be

$$D((t,p),S) := \begin{cases} \min_{q \in S(t)} \|(t,p) - q\| & S(t) \neq \emptyset \\ \infty & \text{otherwise.} \end{cases}$$

The point $q \in S$ is a *projection* of the point $p \in \mathbb{R}^d$ on S if $D(p, S) = D(p, \{q\})$. Since S is affine, the distance is well defined and the projection of p on S exists and is unique. The *projection of a set* $P \subseteq \mathbb{R}^d$ on S is the union of projections of all points in P on S.

For a sequence of k intervals $I = (I_1, \dots, I_k)$ and a sequence $\mathbb{R}^d$ $S = (S_1, \dots, S_k)$ of k j-affine subspaces in $\mathbb{R}^d$ the *cost* of (I, S) with respect to a set of points $P \subseteq \mathbb{R}^d$ is

$$cost(P, I, S) := \sum_{i=1}^{k} \max\{D((t,p), S_i) \mid (t,p) \in P, t \in I_i\}.$$

For a sequence $W = (w_1, \dots, w_k)$ of k non-negative reals the *cost* of (I, S, W) with respect to a set of points $P \subseteq \mathbb{R}^d$ is

$$\mathbf{cost}(P, I, S, W) := \sum_{i=1}^{k} w_i cost(P, I_i, S_i).$$

A *j-flat-mean* of $P \subset \mathbb{R}^d$ is a j-affine subspace S^* of $\mathbb{R}^d$ which minimizes $cost(P, \mathbb{R}, S)$ over every j-affine subspace S of $\mathbb{R}^d$. We denote the *cost* of S^* over $\mathbb{R}$ with respect to P by

$$opt(P, j) := cost(P, \mathbb{R}, S^*).$$

4.2 Coreset Algorithm

We describe an algorithm which gets a set P of points in a j-affine subspace of $\mathbb{R}^d$ and returns a set of sets of points C which complies with Claim 6. The algorithm gets as input an error parameter $\varepsilon > 0$ and a finite set of points $P \subset \mathbb{R}^d$.

4.2.1 Algorithm Overview

We assume that there is a function Basic-Coreset : $2^{\mathbb{R}^d} \times \mathbb{R}$ which receives a set of points P' and an error parameter $\varepsilon' > 0$ and returns new set of points C' such that for every query Q

$$\begin{aligned}(1-\varepsilon)cost(P', \mathbb{R}, Q) &\leq cost(C', \mathbb{R}, Q)\\ &\leq (1+\varepsilon)cost(P', \mathbb{R}, Q')\end{aligned}$$

Our algorithm is divided into 2 steps, in the first step the points are partitioned into m buckets $B_1, \ldots, B_m$ where m is a positive integer and in the second step we perform the following for each $i \in [m]$:

1. Let $p \in B_i$ be a point with the smallest first coordinate in B_i (ties are broken arbitrarily). Add p to the coreset.
2. Run Basic-Coreset$(B_1 \cup \ldots \cup B_{i-1}, \varepsilon)$ and add the output to the coreset.

Now we proceed to describing step-1 and step-2. When describing the steps, we use the following notation: we say that *adding a point* from a set $A \subseteq \mathbb{R}^d$ to a set $B \subseteq \mathbb{R}^d$ is equivalent to performing the following:

1. Picking a point $(t, q) \in A$ s.t. $(x, y) \in A : t \leq x$ for every point (ties broken arbitrarily).
2. Setting $A \leftarrow A \setminus \{(t, q)\}$
3. Setting $B \leftarrow B \cup \{(t, q)\}$

Step-1 Initialize B_1 to be an empty set. Then add points from P to B_1 until either P is empty or $opt(B_1) > 0$. If P is empty finish. Set $m \leftarrow 2$ and repeat until P is empty:

1. Add points from P to B_m while $opt(B_m) < \varepsilon opt(B_1 \cup B_2 \cup \ldots \cup B_{m-1})$.
2. Increase m by one.

Step-2 Initialize the coreset C to be an empty set. For $i \in [m]$ perform the following:

1. Let $p = (t, q) \in B_i$ be a point s.t. $t \leq t'$ for every point $p' = (t', q') \in B_i$ (ties are broken arbitrarily). Set $C \leftarrow C \cup \{p\}$.
2. Run BASIC-CORESET on $B_1 \cup \ldots B_{i-1}$ and ε and add the output to the coreset: $C \leftarrow C \cup \text{BASIC-CORESET}(B_1 \cup \ldots \cup B_{i-1}, \varepsilon)$.

4.2.2 Pseudo-Code

Algorithm 4.1: CREATE-CORESET(ε, P)

comment: $\varepsilon \in \mathbb{R}_{>0}$ is a real and $P \subset \mathbb{R}^d$ is a finite set of points

procedure STEP-1(P, ε)
$B_1 \leftarrow \emptyset$
while $opt(B_0) > 0$ **and** $P \neq \emptyset$
do $\begin{cases} p \leftarrow \text{arbitrary point in } \{(t,p) \in P \mid \forall (t',p') \in P : t \leq t'\} \\ B_1 \leftarrow B_1 \cup \{p\} \\ P \leftarrow P \setminus \{p\} \end{cases}$
$m \leftarrow 1$
while $P \neq \emptyset$
do $\begin{cases} p \leftarrow \text{arbitrary point in } \{(t,p) \in P \mid \forall (t',p') \in P : t \leq t'\} \\ B_m \leftarrow B_m \cup \{p\} \\ P \leftarrow P \setminus \{p\} \end{cases}$
return $(B_1, \ldots, B_m)$

procedure STEP-2$(B_1, \ldots, B_m)$
$C \leftarrow \emptyset$
for $i \leftarrow 1$ **to** m
do $\begin{cases} p \leftarrow \text{arbitrary point in } \{(t,p) \in B_i \mid \forall (t',p') \in B_i : t \leq t'\} \\ C \leftarrow C \cup \{p\} \\ C \leftarrow C \cup \text{BASIC-CORESET}(B_1 \cup \ldots \cup B_{i-1}, \varepsilon). \end{cases}$
return (C)

main
$B_1, \ldots, B_m \leftarrow$ STEP-1(P, ε)
return (STEP-2$(B_1, \ldots, B_m)$)

4.3 Algorithm Guarantees

Theorem 6. *Let $\varepsilon > 0$ be a real number and let P be a set of points in $\mathbb{R}^d$ and let C be the output of Algorithm* **??** *running on P and ε. Let $t \in \mathbb{R}^d$*

and let $I = (-\infty, t]$ be an interval. Then for every point $q \in \mathbb{R}^d$ we have

$$\begin{aligned}(1-\varepsilon)cost(P, I, \{q\}) &\leq cost(C, I, \{q\})\\ &\leq (1+\varepsilon)cost(P, I, \{q\})\end{aligned}$$

Proof. Let $\{B_1, \ldots, B_m\}$ be the partition of P into m buckets as returned by the algorithm. Let $\{I_1, \ldots, I_m\}$ be a partition of $\mathbb{R}$ by set of intervals such that I_i contains the first coordinate of every point in B_i for $i \in [m]$. Let $i \in [m]$ be an integer such that $t \in I_i$. Let $p' \in \mathbb{R}^d$ be a point s.t. $p' \in B_i$ and $p' = (x', y')$ and $x' \leq t$, such a point exists because of algorithm construction.

If there is a point $p \in B_1 \cup \ldots \cup B_{i-1}$ such that $cost(\{p\}, I, \{q\}) = cost(P, I, \{q\})$ then

$$\begin{aligned}(1-\varepsilon)cost(P, I, \{q\}) &\leq cost(C, I, \{q\})\\ &\leq (1+\varepsilon)cost(P, I, \{q\})\end{aligned}$$

because C contains BASIC-CORESET$(B_1 \cup \ldots \cup B_{i-1}, \varepsilon)$.

Otherwise, there must be $p \in B_i$ such that $cost(\{p\}, I, \{q\}) = cost(P, I, \{q\})$. From lemma 7.1 in [?] follows that for every pair of points $a, b, c \in \mathbb{R}^d$, for every $\sigma \in (0, 1)$:

$$\left|D^2(a, c) - D^2(b, c)\right| \leq \frac{12D^2(a, b)}{\sigma} + \frac{\sigma}{2}D^2(a, c).$$

By choosing $\sigma = \sqrt{\varepsilon}$ we get:

$$\left|D^2(q, p) - D^2(q, p')\right| \leq \frac{12D^2(p, p')}{\sqrt{\varepsilon}} + \frac{\sqrt{\varepsilon}}{2}D^2(q, p').$$

From algorithm construction we have:

$$\frac{12D^2(p, p')}{\sqrt{\varepsilon}} \leq \frac{12 \cdot 4\varepsilon cost(C, I, \{q\})}{\sqrt{\varepsilon}}$$

and on the other hand

$$D^2(q, p') \leq cost(C, I, \{q\})$$

since $p' \in C$. Therefore

$$\begin{aligned}\left|D^2(q, p) - D^2(q, p')\right| &\leq 48\sqrt{\varepsilon}cost(C, I, \{q\})\\ &+ \frac{\sqrt{\varepsilon}}{2}cost(C, I, \{q\})\\ &48\sqrt{\varepsilon}cost(C, I, \{q\}).\end{aligned}$$

Hence, we get that

$$
\begin{aligned}
(1-\varepsilon)cost(P, I, \{q\}) &\leq cost(C, I, \{q\}) \\
&\leq (1+\varepsilon)cost(P, I, \{q\})
\end{aligned}
$$

by setting $\varepsilon = (\varepsilon/48)^2$. □

Theorem 7. *For every real $\varepsilon > 0$ and for every positive integer Δ, if a set of n points P is a subset of Δ^d, that is, the points lie on a grid, and if C_0 is the output of* Step-0(P, ε) *then $|C_0| = O(\frac{\log(\Delta dn)}{\varepsilon})$.*

5 2^h-Approximation Of Optimal Decision Trees

Let $A, B \in \mathbb{R}$ be a pair of reals and let $\alpha : [A, B] \times [A, B] \to \mathbb{R}_{\geq 0}$ be a continuous nonnegative function such that if $t_1 \leq t_2 \leq t_3$ then $\alpha(t_1, t_2) + \alpha(t_2, t_3) \leq \alpha(t_1, t_4)$ for every $t_1, t_2, t_3 \in [A, B]$ and $\alpha(t, t) = 0$ for every $t \in [A, B]$.

Let $\beta : [A, B] \times [A, B] \times \mathbb{Z}_{\geq 0} \to \mathbb{R}$ be defined as follows:

$$
\beta(a, b, h) := \begin{cases} \alpha(a, b) & h = 0 \\ \min_{x \in [a,b]} \beta(a, x, h-1) + \beta(x, b, h-1) & h > 0 \end{cases}
$$

Let $\gamma, \delta : [A, B] \times [A, B] \times \mathbb{Z}_{\geq 0} \to \mathbb{R}$ be a pair of mutually recursive functions defined as follows:

$$
\gamma(a, b, h) := \begin{cases} \alpha(a, b) & h = 0 \\ 2\gamma(a, \delta(a, b, h), h-1) & h > 0 \end{cases}
$$

and

$$
\delta(a, b, h) := \min \{x \in [a, b] \mid \gamma(a, x, h-1) = \gamma(x, b, h-1)\}.
$$

Lemma 1. *Let $h \in \mathbb{Z}_{>0}$ be a positive integer. Let $a, b \in [A, B]$ be a pair of real numbers such that $a \leq b$. Then*

$$
\gamma(a, b, h) \leq 2^h \beta(a, b, h)
$$

Proof. We prove by induction on h.

Basis: $h = 0$. Since $\gamma(a, b, 0) = \beta(a, b, 0) = \alpha(a, b)$ the claim follows immediately.

We now assume that the claim is true for $h-1$ and prove it for h. Let $x^* \in [a, b]$ be a real number such that

$$
\beta(a, x^*, h-1) + \beta(x^*, b, h-1) = \beta(a, b, h)
$$

and let $x' \in [a,b]$ be a real number such that

$$\gamma(a, x', h-1) = \gamma(x', b, h-1).$$

We assume that for every $a', b' \in [A, B]$, $a' \leq b'$,

$$\gamma(a', b', h-1) \leq 2^{h-1}\beta(a', b', h-1)$$

and we aim to prove that

$$\gamma(a, b, h) \leq 2^h \beta(a, b, h).$$

We leave the proof that if $t_1 \leq t_2 \leq t_3$ then $\beta(t_1, t_2, h-1) + \beta(t_2, t_3, h-1) \leq \beta(t_1, t_3, h-1)$ for every $t_1, t_2, t_3 \in [A, B]$ to Appendix A.

Let's assume w.l.o.g that $x^* \geq x'$ (the proof is symmetrical for the second case). Then $\beta(a, x^*, h-1) \geq \beta(a, x', h-1)$ and therefore

$$\begin{aligned} 2^{h-1}\beta(a,b,h) &= 2^{h-1}\beta(a, x^*, h-1) + 2^{h-1}\beta(x^*, b, h-1) \\ &\geq 2^{h-1}\beta(a, x^*, h-1) \geq 2^{h-1}\beta(a, x', h-1) \end{aligned}$$

and since we assumed (by induction) that

$$\gamma(a, x', h-1) \leq 2^{h-1}\beta(a, x', h-1)$$

we got that

$$2^{h-1}\beta(a,b,h) \geq 2^{h-1}\beta(a, x', h-1) \geq \gamma(a, x', h-1)$$

and therefore

$$\gamma(a, b, h) = 2\gamma(a, x', h-1) \leq 2^h \beta(a, b, h).$$

□

A Utilities

Lemma 2. *Let $A, B \in \mathbb{R}$ be a pair of reals and let $\alpha : [A, B] \times [A, B] \to \mathbb{R}_{\geq 0}$ be a continuous nonnegative function such that if $t_1 \leq t_2 \leq t_3$ then $\alpha(t_1, t_2) + \alpha(t_2, t_3) \leq \alpha(t_1, t_3)$ for every $t_1, t_2, t_3 \in [A, B]$.*

Let $\beta : [A, B] \times [A, B] \times \mathbb{Z}_{\geq 0} \to \mathbb{R}$ be defined as follows:

$$\beta(a, b, h) := \begin{cases} \alpha(a, b) & h = 0 \\ \min_{x \in [a,b]} \beta(a, x, h-1) + \beta(x, b, h-1) & h > 0 \end{cases}$$

Then for every nonnegative $h \in \mathbb{Z}_{\geq 0}$

$$\beta(t_1, t_2, h) + \beta(t_2, t_3, h) \leq \beta(t_1, t_4, h).$$

for every $t_1, t_2, t_3 \in \mathbb{R}$ such that $t_1 \leq t_2 \leq t_3$

Proof. The proof is by induction on h. Basis: $h = 0$. $\beta(a, b, 0) = \alpha(a, b)$ for every $a, b \in \mathbb{R}$ and the claim follows from the definition of α.

We now assume the claim is correct for $h - 1$ and aim to prove it for h.

Let $x^* \in [A, B]$ be a real such that

$$\beta(t_1, t_3, h) = \beta(t_1, x^*, h - 1) + \beta(x^*, t_3, h - 1),$$

Let us assume w.l.o.g that $x^* \leq t_2$ (the other case is symmetric). Therefore by induction assumption and by the definition of β we get

$$\begin{aligned}
\beta(t_1, t_3, h) &= \beta(t_1, x^*, h - 1) + \beta(x^*, t_3, h - 1) \\
&\geq \beta(t_1, x^*, h - 1) + \beta(x^*, t_2, h - 1) + \beta(t_2, t_3, h - 1) \\
&\geq \beta(t_1, t_2, h) + \beta(t_2, t_3, h - 1) \\
&\geq \beta(t_1, t_2, h) + \beta(t_2, t_3, h).
\end{aligned}$$

□