

Learning Hyperplane Tree: A Piecewise Linear and Interpretable Decision-making Framework (Convergence Proof and Complexity Analysis)

I. CONVERGENCE ANALYSIS OF THE BLOCK SPLITTING PROCESS

In this section, we establish the convergence of the proposed block splitting algorithm. We prove that each split operation results in two non-empty subblocks, ensuring the process terminates.

A. Assumptions

We make the following standard assumptions for the analysis:

Assumption 1. *The block to be split contains at least two samples ($n \geq 2$). Furthermore, the block contains at least one sample belonging to the target class and at least one sample belonging to the non-target class.*

Assumption 2. *The feature-weighted sums $FS(\mathbf{x}_j)$ are distinct for at least two samples within the block. This ensures that a split is always possible.*

Assumption 3. *The threshold γ used in the splitting criterion is a positive integer satisfying $1 \leq \gamma \leq n$.*

B. Notation and Setup

Let a block consist of n samples $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, where each $\mathbf{x}_j \in \mathbb{R}^m$. The feature-weighted sum for sample \mathbf{x}_j is defined as:

$$FS(\mathbf{x}_j) = \sum_{i=1}^m w_i x_{ij},$$

where w_i represents the weight associated with the i -th feature. The block splitting process partitions the samples based on a threshold c . A sample \mathbf{x}_j is assigned to the first subblock if $FS(\mathbf{x}_j) < c$ and to the second subblock if $FS(\mathbf{x}_j) \geq c$.

Let $\mathcal{S} = \{1, \dots, n\}$ be the index set of samples in the block. Let $\mathcal{S}^t \subseteq \mathcal{S}$ be the index set of target class samples and $\mathcal{S}^{nt} \subseteq \mathcal{S}$ be the index set of non-target class samples. By Assumption 1, $\mathcal{S}^t \neq \emptyset$ and $\mathcal{S}^{nt} \neq \emptyset$. Let $\mathcal{FS} = \{FS(\mathbf{x}_j) \mid j \in \mathcal{S}\}$ be the set of feature-weighted sums. We define:

- $\mathcal{FS}^t = \{FS(\mathbf{x}_j) \mid j \in \mathcal{S}^t\}$
- $\mathcal{FS}^{nt} = \{FS(\mathbf{x}_j) \mid j \in \mathcal{S}^{nt}\}$
- $\max TFS = \max(\mathcal{FS}^t)$, $\min TFS = \min(\mathcal{FS}^t)$
- $\max NFS = \max(\mathcal{FS}^{nt})$, $\min NFS = \min(\mathcal{FS}^{nt})$

Note that these extrema exist since \mathcal{FS}^t and \mathcal{FS}^{nt} are non-empty finite sets.

The splitting threshold c is selected based on the following quantities:

$$\begin{aligned} N_1 &= |\{j \in \mathcal{S}^t \mid FS(\mathbf{x}_j) < \min NFS\}|, \\ N_2 &= |\{j \in \mathcal{S}^t \mid FS(\mathbf{x}_j) > \max NFS\}|, \\ N_3 &= |\{j \in \mathcal{S}^{nt} \mid FS(\mathbf{x}_j) < \min TFS\}|, \\ N_4 &= |\{j \in \mathcal{S}^{nt} \mid FS(\mathbf{x}_j) > \max TFS\}|. \end{aligned}$$

Let $N_{\max} = \max\{N_1, N_2, N_3, N_4\}$. The threshold c is chosen as follows:

$$c = \begin{cases} \min NFS, & \text{if } N_1 = N_{\max} \text{ and } N_{\max} \geq \gamma, \\ \max NFS + \delta_1, & \text{if } N_2 = N_{\max} \text{ and } N_{\max} \geq \gamma, \\ \min TFS, & \text{if } N_3 = N_{\max} \text{ and } N_{\max} \geq \gamma, \\ \max TFS + \delta_1, & \text{if } N_4 = N_{\max} \text{ and } N_{\max} \geq \gamma, \\ e, & \text{if } N_{\max} < \gamma, \end{cases}$$

where $e = \frac{\min NFS + \max NFS + \min TFS + \max TFS}{4}$, and δ_1 is an infinitesimally small positive number ensuring that c is strictly greater than $\max NFS$ or $\max TFS$ respectively, but smaller than the next distinct value in \mathcal{FS} if one exists.

C. Convergence Guarantee

We now state and prove the main theorem regarding the convergence of the splitting process.

Theorem 1. *Under Assumptions 1, 2, and 3, the block splitting process based on the threshold c defined above always partitions the block into two non-empty subblocks: $B_1 = \{\mathbf{x}_j \mid \text{FS}(\mathbf{x}_j) < c\}$ and $B_2 = \{\mathbf{x}_j \mid \text{FS}(\mathbf{x}_j) \geq c\}$.*

Proof. We prove the theorem by considering two cases based on the value of c . Let $B_1 = \{\mathbf{x}_j \mid \text{FS}(\mathbf{x}_j) < c\}$ and $B_2 = \{\mathbf{x}_j \mid \text{FS}(\mathbf{x}_j) \geq c\}$. We need to show that $B_1 \neq \emptyset$ and $B_2 \neq \emptyset$.

Case 1: $c \neq e$ In this case, $N_{\max} \geq \gamma \geq 1$. This implies that at least one of N_1, N_2, N_3, N_4 is positive. We analyze the sub-cases based on the choice of c :

- **Sub-case 1.1:** $c = \min \text{NFS}$. Here, $N_1 = N_{\max} \geq \gamma$. By definition, there are $N_1 \geq 1$ target samples \mathbf{x}_j such that $\text{FS}(\mathbf{x}_j) < \min \text{NFS} = c$. These samples belong to B_1 . Thus, $B_1 \neq \emptyset$. Furthermore, all non-target samples \mathbf{x}_k satisfy $\text{FS}(\mathbf{x}_k) \geq \min \text{NFS} = c$. Since the original block contains at least one non-target sample (Assumption 1), these non-target samples belong to B_2 . Thus, $B_2 \neq \emptyset$.
- **Sub-case 1.2:** $c = \max \text{NFS} + \delta_1$. Here, $N_2 = N_{\max} \geq \gamma$. By definition, there are $N_2 \geq 1$ target samples \mathbf{x}_j such that $\text{FS}(\mathbf{x}_j) > \max \text{NFS}$. Since δ_1 is infinitesimally small, $\text{FS}(\mathbf{x}_j) \geq \max \text{NFS} + \delta_1 = c$ holds for these samples (assuming no $\text{FS}(\mathbf{x}_j)$ falls exactly at $\max \text{NFS}$; if it does, the strict inequality $>$ in the definition of N_2 applies. The use of δ_1 ensures the split occurs correctly even with potential ties if handled properly, but the core idea is that these N_2 samples end up in B_2). Thus, $B_2 \neq \emptyset$. All non-target samples \mathbf{x}_k satisfy $\text{FS}(\mathbf{x}_k) \leq \max \text{NFS} < \max \text{NFS} + \delta_1 = c$. Since the block contains at least one non-target sample, these samples belong to B_1 . Thus, $B_1 \neq \emptyset$.
- **Sub-case 1.3:** $c = \min \text{TFS}$. Here, $N_3 = N_{\max} \geq \gamma$. By definition, there are $N_3 \geq 1$ non-target samples \mathbf{x}_j such that $\text{FS}(\mathbf{x}_j) < \min \text{TFS} = c$. These samples belong to B_1 . Thus, $B_1 \neq \emptyset$. All target samples \mathbf{x}_k satisfy $\text{FS}(\mathbf{x}_k) \geq \min \text{TFS} = c$. Since the original block contains at least one target sample (Assumption 1), these target samples belong to B_2 . Thus, $B_2 \neq \emptyset$.
- **Sub-case 1.4:** $c = \max \text{TFS} + \delta_1$. Here, $N_4 = N_{\max} \geq \gamma$. By definition, there are $N_4 \geq 1$ non-target samples \mathbf{x}_j such that $\text{FS}(\mathbf{x}_j) > \max \text{TFS}$. As in Sub-case 1.2, these samples satisfy $\text{FS}(\mathbf{x}_j) \geq \max \text{TFS} + \delta_1 = c$ and belong to B_2 . Thus, $B_2 \neq \emptyset$. All target samples \mathbf{x}_k satisfy $\text{FS}(\mathbf{x}_k) \leq \max \text{TFS} < \max \text{TFS} + \delta_1 = c$. Since the block contains at least one target sample, these samples belong to B_1 . Thus, $B_1 \neq \emptyset$.

In all sub-cases where $c \neq e$, both B_1 and B_2 are non-empty.

Case 2: $c = e$ This case occurs when $N_{\max} < \gamma$. The threshold $c = e$ is the average of the four extreme feature-weighted sum values ($\min \text{NFS}$, $\max \text{NFS}$, $\min \text{TFS}$, $\max \text{TFS}$). Let $\text{FS}_{\min} = \min(\mathcal{FS})$ and $\text{FS}_{\max} = \max(\mathcal{FS})$. By definition, $\min \text{NFS} \geq \text{FS}_{\min}$, $\max \text{NFS} \leq \text{FS}_{\max}$, $\min \text{TFS} \geq \text{FS}_{\min}$, and $\max \text{TFS} \leq \text{FS}_{\max}$. Therefore,

$$\text{FS}_{\min} \leq \frac{\text{FS}_{\min} + \text{FS}_{\min} + \text{FS}_{\min} + \text{FS}_{\min}}{4} \leq e \leq \frac{\text{FS}_{\max} + \text{FS}_{\max} + \text{FS}_{\max} + \text{FS}_{\max}}{4} = \text{FS}_{\max}.$$

So, e lies within the range $[\text{FS}_{\min}, \text{FS}_{\max}]$. By Assumption 2, there exist at least two samples with distinct feature-weighted sums. Thus, $\text{FS}_{\min} < \text{FS}_{\max}$ (since $n \geq 2$). If all $\text{FS}(\mathbf{x}_j)$ were equal, the block would not be splittable, contradicting Assumption 2. Therefore, there must be at least one sample \mathbf{x}_k such that $\text{FS}(\mathbf{x}_k) = \text{FS}_{\min}$ and at least one sample \mathbf{x}_l such that $\text{FS}(\mathbf{x}_l) = \text{FS}_{\max}$.

Can $c = e$ be equal to FS_{\min} or FS_{\max} ? If $e = \text{FS}_{\min}$, then $\min \text{NFS} = \max \text{NFS} = \min \text{TFS} = \max \text{TFS} = \text{FS}_{\min}$. This implies all samples have the same feature-weighted sum FS_{\min} , contradicting Assumption 2. Similarly, $e = \text{FS}_{\max}$ leads to a contradiction. Therefore, $\text{FS}_{\min} < e < \text{FS}_{\max}$.

Since $\text{FS}_{\min} < e$, the sample(s) \mathbf{x}_k with $\text{FS}(\mathbf{x}_k) = \text{FS}_{\min}$ satisfy $\text{FS}(\mathbf{x}_k) < e$. Thus, $B_1 \neq \emptyset$. Since $e < \text{FS}_{\max}$, the sample(s) \mathbf{x}_l with $\text{FS}(\mathbf{x}_l) = \text{FS}_{\max}$ satisfy $\text{FS}(\mathbf{x}_l) > e$, which implies $\text{FS}(\mathbf{x}_l) \geq e$. Thus, $B_2 \neq \emptyset$.

In both Case 1 and Case 2, the splitting process results in two non-empty subblocks B_1 and B_2 . Since each split reduces the size of the block being considered (as $|B_1| \geq 1$, $|B_2| \geq 1$, and $|B_1| + |B_2| = n$), and the process stops when blocks meet certain criteria (e.g., size or purity), the overall block splitting process is guaranteed to converge. \square

II. COMPUTATIONAL COMPLEXITY ANALYSIS

We analyze the time complexity of constructing the LHT.

Theorem 2. *Given a dataset with n samples and m features, the time complexity required to build an LHT up to a depth d is $O(nmd)$.*

Proof. The construction of the LHT involves recursive partitioning of the data at each node. Let's analyze the computational cost at a single node and then aggregate it over the tree structure.

Consider a node containing n_{node} samples. The primary computations performed at this node before splitting are:

- 1) **Feature Expectation/Weight Calculation:** This step typically involves calculating statistics for each feature based on the n_{node} samples within the current node. If we need to iterate through all n_{node} samples for each of the m features (e.g., to compute means or other sums), this requires $O(m \cdot n_{\text{node}})$ time.

2) **Feature-Weighted Sum Calculation:** For each of the n_{node} samples, calculating its feature-weighted sum requires iterating through its m features, involving m multiplications and $m - 1$ additions. This takes $O(m)$ time per sample. For all n_{node} samples at the node, the total time is $O(m \cdot n_{\text{node}})$.

Combining these steps, the total work performed at a single node is dominated by these calculations, resulting in a complexity of $O(m \cdot n_{\text{node}})$ per node.

Now, let's consider the complexity across the entire tree up to depth d . A common way to analyze tree algorithms is level by level. At any given level k (where $0 \leq k < d$), let the nodes be $N_{k,1}, N_{k,2}, \dots$. Let $n_{k,i}$ be the number of samples in node $N_{k,i}$. The crucial observation is that each sample from the original dataset belongs to exactly one node at level k . Therefore, the total number of samples across all nodes at level k is $\sum_i n_{k,i} \leq n$.

The total work performed across all nodes at level k is the sum of the work done at each node:

$$\sum_i O(m \cdot n_{k,i}) = O\left(m \sum_i n_{k,i}\right) = O(mn).$$

This means that the total computational cost for processing all samples across one entire level of the tree is $O(mn)$.

Since the tree is built up to a depth d , there are d such levels (from level 0 to level $d - 1$) where these computations are performed. Therefore, the total time complexity for building the LHT is the cost per level multiplied by the number of levels:

$$d \times O(mn) = O(nmd).$$

This completes the proof. □