Nearly Optimal Bounds for Computing Decision Tree Splits in Data Streams

Hoang Ta[1]
Hanoi University of Science and Technology, Vietnam

Hoa T. Vu[1]
San Diego State University, CA, USA

Abstract

We establish nearly optimal upper and lower bounds for approximating decision tree splits in data streams. For regression with labels in the range $\{0, 1, \ldots, M\}$, we give a one-pass algorithm using $\widetilde{\mathcal{O}}\left(M^2/\varepsilon\right)$ space [2] that outputs a split within additive ε error of the optimal split, improving upon the two-pass algorithm of Pham et al. (ISIT 2025). Furthermore, we provide a matching one-pass lower bound showing that $\Omega(M^2/\varepsilon)$ space is indeed necessary.

For classification, we also obtain a one-pass algorithm using $\widetilde{\mathcal{O}}(1/\varepsilon)$ space for approximating the optimal Gini split, improving upon the previous $\widetilde{\mathcal{O}}\left(1/\varepsilon^2\right)$-space algorithm. We complement these results with matching space lower bounds: $\Omega(1/\varepsilon)$ for Gini impurity and $\Omega(1/\varepsilon)$ for misclassification (which matches the upper bound obtained by sampling).

Our algorithms exploit the Lipschitz property of the loss functions and use reservoir sampling along with Count–Min sketches with range queries. Our lower bounds follow from careful reductions from the INDEX problem.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms; Theory of computation → Lower bounds and information complexity; Theory of computation → Sketching and sampling

Keywords and phrases Decision trees, Streaming algorithms, Lower bounds

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.42

Related Version *Full Version*: `https://arxiv.org/abs/2604.20394`

Funding *Hoa T. Vu*: Supported by NSF Grant No. 2342527.

1 Introduction

Decision trees are ubiquitous in machine learning. They are used as base learners for powerful ensemble methods such as random forests [19], gradient boosting [26, 15], AdaBoost [14], XGBoost [8], LightGBM [24], and CatBoost [30]. These ensemble models achieve state-of-the-art performance on many tasks, particularly on tabular data [17, 33]. At the algorithmic level, a basic primitive in these methods is *split selection*: given a feature and a loss criterion, choose the threshold that yields the best partition of the data [6, 31].

In large-scale settings, the training data may arrive continuously as a stream and cannot be stored in memory in its entirety. This makes it natural to ask whether one can approximate the best split using only one or a few passes and sublinear space. Beyond its practical relevance, this question is also of independent theoretical interest since split selection is the core operation repeated throughout tree learning and other data analysis tools.

[1] Corresponding author.
[2] $\widetilde{\mathcal{O}}(\cdot)$ hides polylogarithmic factors.

In this paper, we study this problem in the insertion-only streaming model and establish nearly optimal bounds for several standard split objectives including mean squared error for regression, misclassification rate, and Gini impurity for classification.

Related Work. Decision trees in data streams have been extensively studied. Domingos and Hulten [11] introduced VFDT, the seminal decision tree learning algorithm for classification for an infinite i.i.d. stream; regression is not addressed in their work. A large subsequent literature has studied adaptive random forests [16], concept drift [36, 7], and other extensions of tree learning in data streams [21, 23, 4, 32, 25]. Others have provided open-source implementations [5, 27] of decision tree learning from data streams, notably the Python packages SCIKIT-MULTIFLOW and RIVER.

Recently, Tatti [35] studied fast algorithms for finding decision tree splits in sparse data streams, Silva et al. [34] considered split computation in federated streaming settings, and Assis et al. [3] investigated split selection from the perspective of structural robustness in evolving streams.

In contrast to the work of Domingos and Hulten [11], which focused only on classification, we do not assume that the elements in the stream are independently and identically distributed. Another challenge we must address is that, unlike misclassification, regression and Gini-type objectives depend on aggregate statistics over ranges of feature values, and these quantities must be estimated accurately in one pass using sublinear space. Our approach combines two ingredients: reservoir sampling, which yields a small candidate set containing a near-optimal split, and coupled dyadic Count–Min sketches, which maintain the range statistics needed to evaluate the loss. For regression, this gives a truly one-pass improvement over the previous two-pass result of [29]; for classification with Gini impurity, a simple label re-encoding reduces the problem to the regression setting.

Since exact decision tree learning is NP-hard [22], greedy heuristics are commonly employed. Top-down frameworks like CART [6] and ID3 [31] recursively optimize feature thresholds to minimize splitting criteria (e.g., Gini impurity and MSE), which serve as the core computational primitives of tree construction.

Problem formulation. We model the feature domain as $\{1, 2, \ldots, N\}$. For real-valued features, this reflects discretization into bins [6, 9, 31, 20, 12, 13].

The following formulation of the optimal split problem follows the standard approach of [6] and [31]. For a more recent overview, we refer to Chapter 9 of [18].

If k is a positive integer, we use $[k]$ to denote the set $\{1, 2, \ldots, k\}$. We denote the indicator variable for the event Z by $\mathbf{1}[Z]$.

Regression. For regression, we are given a stream of pairs

$$(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m), \qquad x_i \in [N],\ y_i \in \{0, 1, 2, \ldots, M\}.$$

We assume $y_i \in \{0, 1, 2, \ldots, M\}$ for some $M \geq 0$, since labels can always be shifted and discretized (as computers have finite precision). We further assume $M = \mathrm{poly}(m)$, so that it can be represented using $O(\log m)$ bits, which is typical in the RAM model. For a split $j \in \{0, 1, \ldots, N\}$, let

$$L_j := [1, j], \qquad R_j := [j + 1, N], \qquad \mu(j) = \frac{\sum_{i=1}^{m} \mathbf{1}[x_i \leq j]\, y_i}{\sum_{i=1}^{m} \mathbf{1}[x_i \leq j]}, \qquad \gamma(j) = \frac{\sum_{i=1}^{m} \mathbf{1}[x_i > j]\, y_i}{\sum_{i=1}^{m} \mathbf{1}[x_i > j]}.$$

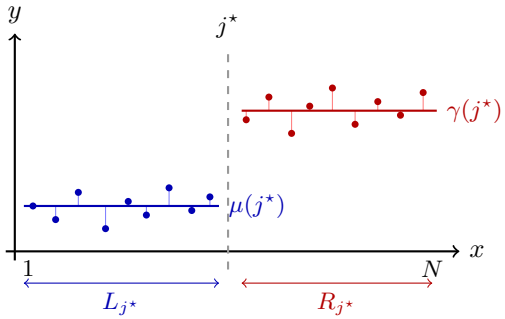

Figure 1 Illustration of the optimal split $j^\star$.

Note that $\mu(j)$ and $\gamma(j)$ are the label means on the left and right sides. They serve as the optimal prediction for $x_i \le j$ and $x_i \ge j+1$ if we split the data at j. The mean squared loss for regression is defined as follows

$$L_{\mathrm{MSE}}(j) = \frac{1}{m}\left(\sum_{i=1}^{m} \mathbf{1}[x_i \le j](y_i - \mu(j))^2 + \sum_{i=1}^{m} \mathbf{1}[x_i > j](y_i - \gamma(j))^2\right). \tag{1}$$

Our goal is to output $\widehat{j}$ such that $L_{\mathrm{MSE}}(\widehat{j}) \le \mathrm{OPT} + \varepsilon$, where

$$\mathrm{OPT} := \min_{0 \le j \le N} L_{\mathrm{MSE}}(j).$$

Classification. For classification, the labels are in $\{-1, +1\}$. The misclassification loss at split j is

$$L_{\mathrm{mis}}(j) = \frac{1}{m}\Big(\min\{f_{-1,[1,j]}, f_{+1,[1,j]}\} + \min\{f_{-1,[j+1,N]}, f_{+1,[j+1,N]}\}\Big).$$

Here, for any interval $R \subseteq [N]$, $f_{+1,R}$ and $f_{-1,R}$ denote the numbers of data points in R with labels $+1$ and -1, respectively. Intuitively, we classify each data point based on the majority on the side of the split. Another popular loss function is the Gini impurity (to be defined formally in Section 2.2):

$$L_{\mathrm{Gini}}(j) = \frac{|L_j|}{m}\,\mathrm{Gini}(\{\{y_i : x_i \le j\}\}) + \frac{|R_j|}{m}\,\mathrm{Gini}(\{\{y_i : x_i \ge j+1\}\}).$$

Motivation. Our objective is to design optimal algorithms that use memory sublinear in m and N and to prove matching lower bounds.

In many large-scale applications, the data does not fit in main memory and must be processed as it arrives while using sublinear space. As more data arrive, m grows and one will run out of main memory (RAM) if we choose to store all the data points.

High-cardinality or composite features (i.e., a feature that is a combination of several features) may also yield a large feature range N, significantly impacting memory and time complexity [13].

Note that even if $O(N)$ space is acceptable, there might be many features and we want to find the best split among all of them. Furthermore, in ensemble methods, we also often train many trees in parallel. Hence, using $o(N)$ space per feature and tree significantly improves the overall memory footprint.

Table 1 Comparison of upper and lower space bounds for numerical split objectives in the streaming model. All guarantees are additive-ε approximations.

Objective	Passes	Space	Reference
Upper bounds			
Regression	2	$\widetilde{\mathcal{O}}\big(M^2/\varepsilon\big)$	[29]
Regression	1	$\widetilde{\mathcal{O}}\big(M^2/\varepsilon\big)$	This Paper
Misclassification	1	$\widetilde{\mathcal{O}}(1/\varepsilon)$	[29]
Gini impurity	1	$\widetilde{\mathcal{O}}\big(1/\varepsilon^2\big)$	[29]
Gini impurity	1	$\widetilde{\mathcal{O}}(1/\varepsilon)$	This Paper
Lower bounds			
Regression	1	$\Omega(M^2/\varepsilon)$	This Paper
Misclassification	1	$\Omega(1/\varepsilon)$	This Paper
Gini impurity	1	$\Omega(1/\varepsilon)$	This Paper

Optimal split selection also arises outside tree learning, including in image segmentation and change-point detection [28, 2].

Providing matching lower and upper bounds is also of independent theoretical interest, as it characterizes the intrinsic difficulty of the problem and helps explain the memory requirements of practical heuristics.

Our results. In this paper, we focus on the one-pass regime. Our main algorithmic contribution is a one-pass additive approximation algorithm for regression using $\widetilde{\mathcal{O}}\big(M^2/\varepsilon\big)$ space, which reduces to $\widetilde{\mathcal{O}}(1/\varepsilon)$ when $M = \mathcal{O}(1)$. This improves upon the two-pass algorithm of [29].

We further show that the same framework yields a one-pass additive approximation algorithm for Gini impurity using $\widetilde{\mathcal{O}}(1/\varepsilon)$ space, improving upon the previous $\widetilde{\mathcal{O}}\big(1/\varepsilon^2\big)$ upper bound in [29].

On the lower-bound side, we prove one-pass space lower bounds for regression, misclassification, and Gini impurity. In particular, our upper and lower bounds are essentially tight, up to polylogarithmic factors, for regression, misclassification, and Gini impurity. Table 1 summarizes the relevant bounds for numerical split objectives.

Our algorithms succeed with high probability, for example, with probability at least $1-1/\mathrm{poly}(N)$ or $1-1/\mathrm{poly}(m)$. Our lower bounds apply to any algorithm that succeeds with probability at least $2/3$. The deterministic complexity of the problem remains an interesting open question.

All omitted proofs can be found in Appendix A.

2 Algorithms

2.1 One-pass additive approximation for regression

In this section we present a truly one-pass additive approximation algorithm for the regression split problem. In contrast with the previous algorithm in [29], the algorithm does not require the stream length m to be known in advance.

At a high level, we obtain a set of candidate splits via reservoir sampling. We show that the loss function satisfies a Lipschitz property which implies that one of the candidates is a good approximation. To this end, the loss of each candidate split is estimated using a coupled dyadic Count–Min sketch that tracks the three range moments required for regression.

The main goal of this section is to prove the following.

▶ **Theorem 1.** *Fix $\varepsilon \in (0,1)$. There exists a randomized one-pass streaming algorithm for the regression split problem that, on every insertion-only stream with labels in $\{0,1,\ldots,M\}$, uses $\widetilde{\mathcal{O}}\left(\frac{M^2}{\varepsilon}\right)$ space and post-processing time, has $\widetilde{\mathcal{O}}(1)$ update time, and with high probability outputs a split $\widehat{j} \in \{0,1,\ldots,N\}$ satisfying $L_{\mathrm{MSE}}(\widehat{j}) \le \mathrm{OPT} + \varepsilon$.*

We first need to define several quantities to be used throughout this section. For every range $R \subseteq [1,N]$, let the count, 1st moment, and 2nd moment be defined as

$$n_R := \sum_{i=1}^{m} \mathbf{1}[x_i \in R], \qquad s_R := \sum_{i=1}^{m} \mathbf{1}[x_i \in R]\, y_i, \qquad q_R := \sum_{i=1}^{m} \mathbf{1}[x_i \in R]\, y_i^2.$$

For every nonempty range R, let $\bar{y}_R := \frac{s_R}{n_R}$. Observe that

$$\sum_{i:x_i \in R} (y_i - \bar{y}_R)^2 = q_R - 2\bar{y}_R s_R + n_R \bar{y}_R^2 = q_R - \frac{2s_R^2}{n_R} + \frac{s_R^2}{n_R} = q_R - \frac{s_R^2}{n_R}.$$

We define the sum squared error of a set as follows.

$$\mathrm{SSE}(R) := \begin{cases} q_R - \dfrac{s_R^2}{n_R}, & n_R > 0, \\ 0, & n_R = 0. \end{cases}$$

For every split j, define $L_j := \{1,2,\ldots,j\}$ and $R_j := \{j+1, j+2, \ldots, N\}$. We have

$$m L_{\mathrm{MSE}}(j) = \mathrm{SSE}(L_j) + \mathrm{SSE}(R_j). \tag{2}$$

Throughout this section, we will use the following quantities:

$$\tau := \frac{\varepsilon}{16M^2}, \qquad \beta := \frac{\varepsilon}{32M^2}, \qquad K := \Theta\left(\frac{\log N}{\tau}\right).$$

We rely on the classic Count–Min sketch with range queries due to Cormode and Muthukrishnan [10].

▶ **Theorem 2** (Count–Min sketch with range queries)**.** *Consider an insertion-only stream of tuples $(x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m)$, where $x_i \in [N]$ and $y_i \in \{0,1,\ldots,M\}$. For any interval $R = [l,r] \subseteq [N]$, define*

$$f_R := \sum_{i:\, x_i \in R} y_i, \qquad \text{and} \qquad W := \sum_{i=1}^{m} y_i.$$

Then there exists a data structure using $\widetilde{\mathcal{O}}(k)$ bits of space and $\widetilde{\mathcal{O}}(1)$ update time such that, with high probability, for all intervals $R = [l,r] \subseteq [N]$, it returns an estimate $\widehat{f}_R$ satisfying

$$f_R \le \widehat{f}_R \le f_R + \frac{W}{k}.$$

We outline our algorithm below.

Algorithm for estimating regression split.

1. During the stream, maintain a reservoir sample of size K, and after the stream define

$$S := \{0, N\} \cup \{x,\ x-1 : (x,y) \text{ appears in the reservoir and } x > 1\}.$$

2. Simultaneously, maintain three Count–Min range-query sketches with $k = 1/\beta$ for the streams

$$\{(x_i, 1)\}_{i\in[m]}, \qquad \{(x_i, y_i)\}_{i\in[m]}, \qquad \{(x_i, y_i^2)\}_{i\in[m]}.$$

That is when (x_i, y_i) arrives, we insert $(x_i, 1)$ into the first sketch, (x_i, y_i) into the second, and (x_i, y_i^2) into the third.

3. For any interval $R \subseteq [N]$, querying the sketches returns estimates $\widehat{n}_R, \widehat{s}_R, \widehat{q}_R$. Set

$$\widehat{\mathrm{SSE}}(R) := \begin{cases} \widehat{q}_R - \dfrac{\widehat{s}_R^2}{\widehat{n}_R + \beta m}, & \widehat{n}_R > 0, \\ 0, & \widehat{n}_R = 0. \end{cases}$$

4. For every candidate split $j \in S$, define

$$\widehat{L}_{\mathrm{MSE}}(j) := \frac{1}{m}\Big(\widehat{\mathrm{SSE}}([1,j]) + \widehat{\mathrm{SSE}}([j+1,N])\Big).$$

5. Output $\widehat{j} := \arg\min_{j\in S} \widehat{L}_{\mathrm{MSE}}(j)$.

▶ **Lemma 3.** *With high probability, the following holds simultaneously for all intervals $R \subseteq [N]$:*

$$\widehat{n}_R = n_R + a_R, \qquad \widehat{s}_R = s_R + b_R, \qquad \widehat{q}_R = q_R + c_R,$$

where

$$0 \le a_R \le \beta m, \qquad 0 \le b_R \le \beta M m, \qquad 0 \le c_R \le \beta M^2 m.$$

Proof. Apply Theorem 2 with $k = \Theta(1/\beta)$ separately to the three insertion-only streams

$$\{(x_i, 1)\}_{i=1}^m, \qquad \{(x_i, y_i)\}_{i=1}^m, \qquad \{(x_i, y_i^2)\}_{i=1}^m.$$

For the first stream, the total weight is m. For the second stream, the total weight is at most mM. Finally, for the third stream, the total weight is at most mM^2. The claim follows. ◀

▶ **Lemma 4.** *Condition on the event in Lemma 3. Then for every candidate split $j \in S$,*

$$|\widehat{L}_{\mathrm{MSE}}(j) - L_{\mathrm{MSE}}(j)| \le \frac{\varepsilon}{4}.$$

Proof. Fix $j \in S$ and let R be either L_j or R_j. Set $D := \widehat{n}_R + \beta m = n_R + a_R + \beta m$, so $D \ge n_R$ and $D \ge \beta m$.

If $n_R = 0$, then $s_R = q_R = 0$ and $\mathrm{SSE}(R) = 0$. If $\widehat{n}_R = 0$, then $\widehat{\mathrm{SSE}}(R) = 0 = \mathrm{SSE}(R)$, so the claim is immediate. Otherwise, if $\widehat{n}_R > 0$, then since $s_R = q_R = 0$, Lemma 3 gives $\widehat{q}_R = q_R + c_R = c_R$ and $\widehat{s}_R = s_R + b_R = b_R$; substituting into $\widehat{\mathrm{SSE}}(R) = \widehat{q}_R - \widehat{s}_R^2/D$ and using $\mathrm{SSE}(R) = 0$ yields

$$\widehat{\mathrm{SSE}}(R) - \mathrm{SSE}(R) = c_R - \frac{b_R^2}{D}.$$

Since $c_R \geq 0$ we get $\widehat{\mathrm{SSE}}(R) - \mathrm{SSE}(R) \leq c_R \leq \beta M^2 m$, and since $D \geq \beta m$ we get

$$\widehat{\mathrm{SSE}}(R) - \mathrm{SSE}(R) \geq -\frac{b_R^2}{D} \geq -\frac{(\beta M m)^2}{\beta m} = -\beta M^2 m.$$

Hence $|\widehat{\mathrm{SSE}}(R) - \mathrm{SSE}(R)| \leq \beta M^2 m$.

Henceforth assume $n_R \geq 1$. Observe that

$$\widehat{\mathrm{SSE}}(R) - \mathrm{SSE}(R) = \left(q_R + c_R - \frac{(s_R + b_R)^2}{D}\right) - \left(q_R - \frac{s_R^2}{n_R}\right) = c_R + \frac{s_R^2}{n_R} - \frac{(s_R + b_R)^2}{D}.$$

We can upper bound the difference as follows:

$$\begin{aligned}
\widehat{\mathrm{SSE}}(R) - \mathrm{SSE}(R) &= c_R + \frac{s_R^2}{n_R} - \frac{(s_R + b_R)^2}{D} \\
&\leq c_R + \frac{s_R^2}{n_R} - \frac{s_R^2}{D} && (\text{drop } -2s_R b_R/D - b_R^2/D \leq 0) \\
&= c_R + \frac{s_R^2(D - n_R)}{n_R D} \\
&\leq \beta M^2 m + \frac{M^2 n_R^2 \cdot 2\beta m}{n_R D} && (s_R \leq M n_R, \text{ and } D - n_R \leq 2\beta m) \\
&\leq 3\beta M^2 m. && (n_R/D \leq 1)
\end{aligned}$$

We now lower bound the difference:

$$\begin{aligned}
\widehat{\mathrm{SSE}}(R) - \mathrm{SSE}(R) &= c_R + \frac{s_R^2}{n_R} - \frac{(s_R + b_R)^2}{D} \\
&\geq -\frac{2s_R b_R}{D} - \frac{b_R^2}{D} && (\text{drop } c_R \geq 0 \text{ and } s_R^2/n_R \geq s_R^2/D \geq 0) \\
&\geq -\frac{2M n_R \cdot \beta M m}{n_R} - \frac{(\beta M m)^2}{\beta m} && (D \geq n_R; \text{ and } D \geq \beta m) \\
&= -2\beta M^2 m - \beta M^2 m = -3\beta M^2 m.
\end{aligned}$$

Hence $|\widehat{\mathrm{SSE}}(R) - \mathrm{SSE}(R)| \leq 3\beta M^2 m$. Applying this to both sides and using $m L_{\mathrm{MSE}}(j) = \mathrm{SSE}(L_j) + \mathrm{SSE}(R_j)$,

$$|\widehat{L}_{\mathrm{MSE}}(j) - L_{\mathrm{MSE}}(j)| \leq \frac{6\beta M^2 m}{m} = 6\beta M^2 = \frac{6M^2\varepsilon}{32M^2} = \frac{3\varepsilon}{16} < \frac{\varepsilon}{4}.$$ ◀

The next lemma is from Pham et al. [29]. We include the proof, simplified and with the constant 4 removed, in the appendix for completeness. It bounds the difference between the squared loss at splits j' and j based on the number of data points in $(j, j']$.

▶ **Lemma 5.** *Let $0 \leq j < j' \leq N$, and let $b := \left|\{i : \ j < x_i \leq j'\}\right|$. Then*

$$|L_{\mathrm{MSE}}(j') - L_{\mathrm{MSE}}(j)| \leq \frac{bM^2}{m}.$$

The next lemma is also adapted from [29]. Roughly speaking, if we sample $\approx CM^2/\varepsilon \cdot \log N$ data points for some sufficiently large constant C, then the probability that an interval with at least $\varepsilon m/M^2$ data points has no data point in the sample is upper bounded by

$$\approx \left(1 - \frac{CM^2 \log N}{m\varepsilon}\right)^{\varepsilon m/M^2} \leq e^{-\frac{\varepsilon m}{M^2} \cdot \frac{CM^2 \log N}{m\varepsilon}} \leq \frac{1}{N^4}.$$

Hence, taking a union bound over $\binom{N}{2}$ intervals, we deduce that this could not happen with high probability. This implies that one of the candidate splits must be a good approximation.

▶ **Lemma 6.** *With probability at least $1 - 1/N^2$, every interval $(a, b] \subseteq [N]$ containing more than τm stream items contains at least one sampled item from the final reservoir of size $K = \lceil C \log N/\tau \rceil$, for a sufficiently large constant C.*

▶ **Lemma 7.** *With probability at least $1 - 1/N^2$, there exists a candidate $j \in S$ such that*

$$L_{\mathrm{MSE}}(j) \le \mathrm{OPT} + \frac{\varepsilon}{16}.$$

Proof. Let $j^\star$ be an optimal split, so $L_{\mathrm{MSE}}(j^\star) = \mathrm{OPT}$. Condition on the event in Lemma 6.

If $j^\star \in S$, then we are done. Otherwise, since $N \in S$, the set $\{t \in S : t > j^\star\}$ is nonempty. Let

$$j := \min\{t \in S : t > j^\star\}.$$

We claim that the interval $(j^\star, j]$ contains no sampled feature value.

Indeed, if some sampled point had feature value $x \in (j^\star, j)$, then $x \in S$, contradicting the minimality of j. If some sampled point had feature value $x = j$, then $x - 1 \in S$ by the definition of S, and since $x - 1 \ge j^\star$, this again contradicts the minimality of j unless $x - 1 = j^\star$, in which case $j^\star \in S$, contradiction.

Hence $(j^\star, j]$ contains no sampled point. By the event in Lemma 6, this interval must therefore contain at most τm stream items. Applying Lemma 5,

$$L_{\mathrm{MSE}}(j) - L_{\mathrm{MSE}}(j^\star) \le \frac{\tau m \cdot M^2}{m} = \tau M^2 = \frac{\varepsilon}{16}.$$

Since $L_{\mathrm{MSE}}(j^\star) = \mathrm{OPT}$, we conclude that

$$L_{\mathrm{MSE}}(j) \le \mathrm{OPT} + \frac{\varepsilon}{16}. \qquad \blacktriangleleft$$

We now put it all together to prove Theorem 1.

Proof of Theorem 1. If $M^2 \le \varepsilon/16$ then we output $\widehat{j} = 0$; the average squared loss is at most $M^2 \le \varepsilon/16 < \varepsilon$. Henceforth assume $M^2 > \varepsilon/16$.

The algorithm maintains a reservoir sample of size K and range-query Count–Min sketches with parameter $\beta = \Theta(\varepsilon/M^2)$. The reservoir has $O(1)$ update time and the sketch has $\widetilde{\mathcal{O}}(1)$ update time.

By Lemma 7, with probability at least $1 - 1/N^2$, there exists $j \in S$ such that

$$L_{\mathrm{MSE}}(j) \le \mathrm{OPT} + \frac{\varepsilon}{16}.$$

By Lemma 4, with probability at least $1 - 1/N^2$, every candidate satisfies $|\widehat{L}_{\mathrm{MSE}}(j) - L_{\mathrm{MSE}}(j)| \le \varepsilon/4$. Hence both events hold simultaneously with high probability. Since $\widehat{j}$ minimizes $\widehat{L}_{\mathrm{MSE}}$ over S,

$$L_{\mathrm{MSE}}(\widehat{j}) \le \widehat{L}_{\mathrm{MSE}}(\widehat{j}) + \frac{\varepsilon}{4} \le \widehat{L}_{\mathrm{MSE}}(j) + \frac{\varepsilon}{4} \le L_{\mathrm{MSE}}(j) + \frac{\varepsilon}{2} \le \mathrm{OPT} + \frac{\varepsilon}{16} + \frac{\varepsilon}{2} < \mathrm{OPT} + \varepsilon. \blacktriangleleft$$

2.2 One-pass additive approximation for classification with Gini impurity

In this section, we improve the $\widetilde{\mathcal{O}}\big(1/\varepsilon^2\big)$-space algorithm of [29] to $\widetilde{\mathcal{O}}(1/\varepsilon)$ to match our lower bound in the next section. For a split j, write

$$a = f_{+1,[1,j]}, \qquad b = f_{-1,[1,j]}, \qquad c = f_{+1,[j+1,N]}, \qquad d = f_{-1,[j+1,N]}.$$

The Gini impurity of a multiset S of labels is $1 - \sum_y \left(\frac{|S_y|}{|S|}\right)^2$, where $S_y = \{\{s \in S : s = y\}\}$. We adopt the convention $\mathrm{Gini}(\emptyset) = 0$. For example, if all labels are identical then the impurity is $1 - 1^2 = 0$ (a pure node), whereas if the labels are split evenly between $+1$ and -1 the impurity is $1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 = \frac{1}{2}$, which is the maximum.

For binary labels this equals $\frac{2pq}{(p+q)^2}$ where $p = |S_{+1}|$ and $q = |S_{-1}|$. The Gini loss of split j is the weighted sum of impurities of the two children:

$$L_{\mathrm{Gini}}(j) = \frac{a+b}{m} \cdot \frac{2ab}{(a+b)^2} + \frac{c+d}{m} \cdot \frac{2cd}{(c+d)^2} = \frac{2ab}{m(a+b)} + \frac{2cd}{m(c+d)}.$$

▶ **Lemma 8.** *For any splits $j < j'$ in $[N]$, let $\ell = \left|\{i : j < x_i \le j'\}\right|$ be the number of data points in the interval $(j, j']$. Then*

$$\left|L_{Gini}(j) - L_{Gini}(j')\right| \le \frac{2\ell}{m}.$$

Proof sketch. We re-encode each label $y_i \in \{-1, +1\}$ as $z_i = (y_i+1)/2 \in \{0, 1\}$ and establish the identity $2L_{\mathrm{MSE}}(j) = L_{\mathrm{Gini}}(j)$. The Lipschitz bound then follows by applying Lemma 5 with $M = 1$. ◀

▶ **Theorem 9.** *Fix $\varepsilon \in (0, 1)$. There exists a randomized one-pass streaming algorithm for the Gini split problem that uses $\widetilde{\mathcal{O}}(1/\varepsilon)$ space, has $\widetilde{\mathcal{O}}(1)$ update time, and with high probability outputs $\widehat{j} \in \{0, \ldots, N\}$ satisfying $L_{\mathrm{Gini}}(\widehat{j}) \le \mathrm{OPT}_{\mathrm{Gini}} + \varepsilon$.*

Proof. Re-encode each label $y_i \in \{-1, +1\}$ as $z_i = (y_i + 1)/2 \in \{0, 1\}$. As shown in the proof of Lemma 8, for every split j,

$$L_{\mathrm{Gini}}(j) = 2\, L_{\mathrm{MSE}}(j),$$

where L_{MSE} is the standard mean squared loss on the re-encoded stream. In particular, $\mathrm{OPT}_{\mathrm{Gini}} = 2\ \mathrm{OPT}_{\mathrm{MSE}}$ and both objectives share the same optimal split.

Apply the algorithm of Theorem 1 to the stream (x_i, z_i) with $M = 1$ and target error $\varepsilon/2$. It uses $\widetilde{\mathcal{O}}\big(M^2/(\varepsilon/2)\big) = \widetilde{\mathcal{O}}(1/\varepsilon)$ space and, with high probability, outputs $\widehat{j}$ satisfying

$$L_{\mathrm{MSE}}(\widehat{j}) \le \mathrm{OPT}_{\mathrm{MSE}} + \frac{\varepsilon}{2}.$$

Multiplying through by 2:

$$L_{\mathrm{Gini}}(\widehat{j}) = 2\, L_{\mathrm{MSE}}(\widehat{j}) \le 2\ \mathrm{OPT}_{\mathrm{MSE}} + \varepsilon = \mathrm{OPT}_{\mathrm{Gini}} + \varepsilon.$$ ◀

3 Lower bounds

In this section, we establish one-pass space lower bounds for approximating regression and classification optimal splits. The lower bounds for numerical observations all follow the same high-level template: we reduce from Index by constructing a stream with two competing adjacent candidate splits whose identity reveals the hidden bit.

▶ **Lemma 10** (Index one-way lower bound [1])**.** *In the one-way randomized communication problem Index, Alice holds $z \in \{0, 1\}^n$ and Bob holds $i \in [n]$; Bob must output z_i. Any one-way randomized protocol that succeeds with probability at least $2/3$ requires $\Omega(n)$ bits of communication.*

The main difference between the proofs lies in the loss-specific gap calculation: quadratic for regression, direct majority/minority counting for misclassification.

3.1 Lower bound for regression

Fix $\varepsilon \in (0, 10^{-3})$. Consider the problem of finding the optimal regression split with labels $y_i \in [0, 1]$ (i.e., $M = 1$) and feature values $x_i \in [N]$. Our goal is to show that any one-pass randomized streaming algorithm that, with probability at least $2/3$, outputs a split $\widehat{j} \in [N]$ satisfying $L_{\mathrm{MSE}}(\widehat{j}) \leq \mathrm{OPT} + \varepsilon$ must use $\Omega(1/\varepsilon)$ bits of memory, even for instances with $N = \Theta(1/\varepsilon)$.

The following lemma is well-known. It shows that given a multiset S of numbers, then $\sum_{x \in S}(x - d)^2$ is minimized when d is the average of elements in S.

▶ **Lemma 11.** *Let $y_1, \ldots, y_k \in \mathbb{R}$ and let $\bar{y} := \frac{1}{k}\sum_{t=1}^{k} y_t$. Then for any $a \in \mathbb{R}$,*

$$\sum_{t=1}^{k}(y_t - a)^2 = \sum_{t=1}^{k}(y_t - \bar{y})^2 \; + \; k(a - \bar{y})^2.$$

In particular, $\sum_{t=1}^{k}(y_t - \bar{y})^2 \leq \sum_{t=1}^{k}(y_t - a)^2$ for all a.

The next lemma quantifies how much the squared error objective changes when we append a uniform block of labels to an existing set.

▶ **Lemma 12.** *Let D be a size-a multiset of reals with mean μ, and define $\mathrm{SSE}(D) := \sum_{y \in D}(y - \mu)^2$. Let G be a multiset of size B whose elements are all equal to the same value v (so $\mathrm{SSE}(G) = 0$). Then*

$$\mathrm{SSE}(D \cup G) = \mathrm{SSE}(D) \; + \; \frac{aB}{a + B}\,(\mu - v)^2.$$

▶ **Theorem 13.** *For all sufficiently small $\varepsilon > 0$, the following holds. Consider the optimal regression split problem with labels $y_t \in [0, M]$ and feature values $x_t \in [N]$. Any one-pass randomized streaming algorithm that, with probability at least $2/3$, outputs a split $\widehat{j} \in [N]$ satisfying*

$$L_{\mathrm{MSE}}(\widehat{j}) \leq \mathrm{OPT} + \varepsilon$$

must use $\Omega(M^2/\varepsilon)$ bits of memory.

Proof. We first remove M by scaling. Given labels $y_t \in [0, M]$, define $\tilde{y}_t := y_t/M \in [0, 1]$. For every split j,

$$L_{\mathrm{MSE}}(j;\, y) = M^2\, L_{\mathrm{MSE}}(j;\, \tilde{y}).$$

Therefore, an ε-additive algorithm for labels in $[0, M]$ would give an (ε/M^2)-additive algorithm for labels in $[0, 1]$. So it suffices to prove the theorem for $M = 1$ with target additive error:

$$\varepsilon' := \varepsilon/M^2.$$

We may assume $\varepsilon' \leq 10^{-3}$.

We reduce from Index. Alice encodes her input as a data stream and runs the algorithm on it, then sends the algorithm's memory state to Bob. Bob appends his part of the stream and uses the algorithm's output to determine whether the queried bit is 0 or 1. Let

$$n := \left\lfloor \frac{1}{1000\varepsilon'} \right\rfloor, \qquad B := n, \qquad T := 100n^2, \qquad N := 2n + 1.$$

Alice and Bob will build a stream over labels in $\{0, 1\}$.

Alice's prefix. Given $z \in \{0,1\}^n$, for each $k \in [n]$, Alice inserts B copies of

$$(x, y) = (2k, z_k).$$

Thus Alice contributes $m_A = nB = n^2$ points.

Bob's suffix. Given $i \in [n]$, Bob appends

$$T \text{ copies of } (2i-1, 0) \qquad \text{and} \qquad T \text{ copies of } (2i+1, 1).$$

The total stream length is

$$m = m_A + 2T = n^2 + 200n^2 = 201n^2.$$

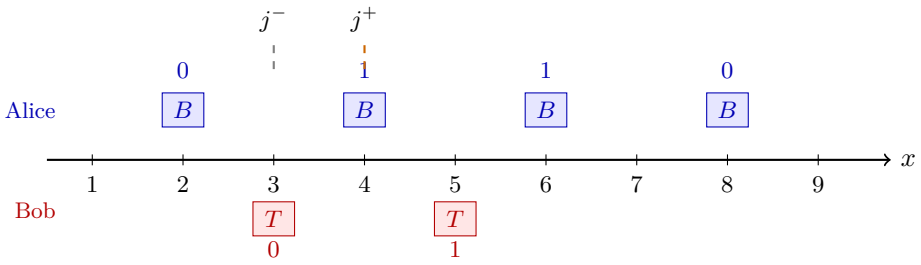

Figure 2 Construction for $n = 4$, $z = (0, 1, 1, 0)$, and $i = 2$. Blue (above): Alice's B copies of each of $(2k, z_k)$. Red (below): Bob's T copies of $(3, 0)$ and $(5, 1)$. Dashed lines mark the two candidate splits $j^- = 3$ and $j^+ = 4$.

Consider the following two splits:

$$j^- := 2i - 1, \qquad j^+ := 2i.$$

These are the two adjacent splits around the block at $x = 2i$.

We first claim that any split that is not j^- or j^+ has high squared error.

▷ Claim 14. For every $j \notin \{j^-, j^+\}$, we have

$$L_{\mathrm{MSE}}(j) \geq \frac{50}{201}.$$

Proof. Fix $j \notin \{j^-, j^+\}$. If $j \leq 2i - 2$, then both anchor blocks, namely the T copies of $(2i-1, 0)$ and the T copies of $(2i+1, 1)$, lie on the right side of the split. Let c be the optimal constant on the right side, i.e., the average label on that side. Then the contribution of the anchor blocks alone is

$$T(0-c)^2 + T(1-c)^2.$$

This is minimized at $c = \frac{1}{2}$, so

$$T(0-c)^2 + T(1-c)^2 \geq T\left(0 - \tfrac{1}{2}\right)^2 + T\left(1 - \tfrac{1}{2}\right)^2 = \frac{T}{2}.$$

If $j \geq 2i + 1$, then both anchor blocks lie on the left side of the split, and the same argument shows that the anchor contribution on the left is at least $T/2$.

Thus, for every $j \notin \{j^-, j^+\}$, the total squared error is at least $T/2$. Since $m = 201n^2$ and $T = 100n^2$, we obtain

$$L_{\mathrm{MSE}}(j) \geq \frac{T/2}{m} = \frac{100n^2/2}{201n^2} = \frac{50}{201}.$$

◁

We now show that both j^- and j^+ have much lower squared error. Hence, the optimal split must be among them.

▷ Claim 15. For $j \in \{j^-, j^+\}$, we have $L_{\text{MSE}}(j) \leq \frac{1}{201}$.

Proof. Recall that j^- splits the feature values into intervals $[1, 2i-1]$ and $[2i, N]$; on the other hand, j^+ splits the feature values into intervals $[1, 2i]$ and $[2i+1, N]$.

In both cases, all T zeros at $x = 2i-1$ lie on the left of j and all T ones at $x = 2i+1$ lie on the right. Using constant 0 on the left and constant 1 on the right, both anchor blocks incur zero error. Only Alice's points contribute, and each contributes at most 1, so

$$L_{\text{MSE}}(j) \leq \frac{m_A}{m} = \frac{n^2}{201n^2} = \frac{1}{201}. \qquad \triangleleft$$

Since $\frac{1}{201} < \frac{50}{201}$, we must have

$$\text{OPT} = \min\{L_{\text{MSE}}(j^-), L_{\text{MSE}}(j^+)\}.$$

The better split between j^- and j^+ reveals z_i. Let $v := z_i \in \{0, 1\}$. Write

$$D_L := \{t : x_t < 2i\}, \qquad D_R := \{t : x_t > 2i\},$$

and let

$$a := |D_L|, \quad \mu := \text{mean label on } D_L, \qquad b := |D_R|, \quad \gamma := \text{mean label on } D_R.$$

Let G be the block of B copies of label v at $x = 2i$. Under j^+, the block G is placed on the left, and under j^-, it is placed on the right.

Because D_L contains exactly $T = 100n^2$ zeros from Bob and at most n^2 ones from Alice,

$$\mu \leq \frac{n^2}{100n^2 + n^2} = \frac{1}{101} < \frac{1}{100}.$$

Similarly, D_R contains exactly $100n^2$ ones from Bob and at most n^2 zeros from Alice, so

$$\gamma \geq \frac{100n^2}{100n^2 + n^2} = \frac{100}{101} > \frac{99}{100}.$$

We note that the function $\frac{a}{a+B}$ is increasing with respect to a. Since $a, b \geq T = 100n^2$ and $B = n$, we have

$$\frac{a}{a+B} \geq \frac{T}{T+B} = \frac{100n^2}{100n^2 + n} > \frac{1}{2},$$

Similarly,

$$\frac{b}{b+B} > \frac{1}{2}.$$

Under the split j^+, the block G is appended to the left side D_L; under the split j^-, it is appended to the right side D_R. Applying Lemma 12 to each case,

$$L_{\text{MSE}}(j^+) = \frac{1}{m}\left[\text{SSE}(D_L) + \frac{aB}{a+B}(\mu - v)^2 + \text{SSE}(D_R)\right],$$
$$L_{\text{MSE}}(j^-) = \frac{1}{m}\left[\text{SSE}(D_L) + \text{SSE}(D_R) + \frac{bB}{b+B}(\gamma - v)^2\right].$$

This gives

$$L_{\text{MSE}}(j^+) - L_{\text{MSE}}(j^-) = \frac{B}{m}\left[\frac{a}{a+B}(\mu - v)^2 - \frac{b}{b+B}(\gamma - v)^2\right].$$

Case 1. If $v = 1$, then

$$(\mu - 1)^2 \geq (99/100)^2, \qquad (\gamma - 1)^2 \leq (1/100)^2.$$

Hence,

$$L_{\mathrm{MSE}}(j^+) - L_{\mathrm{MSE}}(j^-) \geq \frac{B}{m}\left[\frac{1}{2}\left(\frac{99}{100}\right)^2 - \left(\frac{1}{100}\right)^2\right] > \frac{B}{4m} = \frac{1}{804\,n} \implies L_{\mathrm{MSE}}(j^-) < L_{\mathrm{MSE}}(j^+).$$

Case 2. If $v = 0$, then

$$\mu^2 \leq (1/100)^2, \qquad \gamma^2 \geq (99/100)^2.$$

Hence,

$$L_{\mathrm{MSE}}(j^+) - L_{\mathrm{MSE}}(j^-) \leq \frac{B}{m}\left[\left(\frac{1}{100}\right)^2 - \frac{1}{2}\left(\frac{99}{100}\right)^2\right] < -\frac{B}{4m} = -\frac{1}{804\,n} \implies L_{\mathrm{MSE}}(j^+) < L_{\mathrm{MSE}}(j^-).$$

In both cases,

$$|L_{\mathrm{MSE}}(j^+) - L_{\mathrm{MSE}}(j^-)| > \frac{1}{804\,n}.$$

Since

$$n = \left\lfloor \frac{1}{1000\varepsilon'} \right\rfloor, \qquad \text{hence} \qquad \frac{1}{n} \geq 1000\varepsilon',$$

we get

$$|L_{\mathrm{MSE}}(j^+) - L_{\mathrm{MSE}}(j^-)| > \varepsilon'.$$

Therefore any algorithm that outputs $\widehat{j}$ with

$$L_{\mathrm{MSE}}(\widehat{j}) \leq \mathrm{OPT} + \varepsilon'$$

must return the correct minimizer in $\{j^-, j^+\}$, and hence reveals z_i.

So Alice can run the streaming algorithm on her prefix, send its memory state to Bob, and Bob can recover z_i with probability at least $2/3$. This gives a one-way protocol for Index with communication equal to the memory used. By Lemma 10, that memory must be $\Omega(n)$ bits. Finally,

$$n = \Theta(1/\varepsilon') = \Theta(M^2/\varepsilon),$$

so the space lower bound is $\Omega(M^2/\varepsilon)$. ◀

3.2 Lower bound for classification

For classification with numerical observations, we will show the following. Fix $\varepsilon \in (0, 10^{-3})$. Any one-pass randomized streaming algorithm that, with probability at least $2/3$, outputs a split $\widehat{j} \in [N]$ satisfying $L_{\mathrm{mis}}(\widehat{j}) \leq \mathrm{OPT} + \varepsilon$ must use $\Omega(1/\varepsilon)$ bits of memory, even for instances with $N = \Theta(1/\varepsilon)$.

For an interval $R \subseteq [N]$, let

$$f_{+1,R} = |\{i : x_i \in R,\ y_i = +1\}|, \qquad f_{-1,R} = |\{i : x_i \in R,\ y_i = -1\}|$$

denote the number of $+1$ and -1 labels falling in R, respectively.

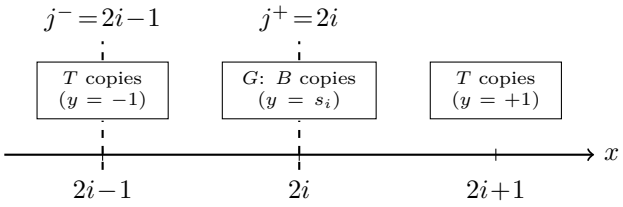

Figure 3 Bob contributes two large anchor blocks at $x = 2i - 1$ (label -1) and $x = 2i + 1$ (label $+1$), while Alice contributes the middle block G at $x = 2i$ with label $s_i \in \{-1, +1\}$. The competitive splits are the two adjacent candidates j^- and j^+; knowing the better split reveals the hidden bit z_i.

▶ **Theorem 16.** *Fix $\varepsilon \in (0, 10^{-3})$. Any one-pass randomized streaming algorithm that, with probability at least $2/3$, outputs a split $\widehat{j} \in [N]$ satisfying*

$$L_{\text{mis}}(\widehat{j}) \leq \text{OPT} + \varepsilon$$

must use $\Omega(1/\varepsilon)$ bits of memory, even for instances with $N = \Theta(1/\varepsilon)$.

Proof. We again reduce from Index (Lemma 10).

$$n = \left\lfloor \frac{1}{100\varepsilon} \right\rfloor, \qquad N = 2n + 1, \qquad B = n, \qquad T = 4n^2.$$

We will construct a stream over $x \in [N]$ with labels in $\{-1, +1\}$.

Alice's prefix. Given $z \in \{0,1\}^n$, for each $k \in [n]$ Alice inserts exactly B copies of

$$(x, y) = (2k,\ s_k), \qquad \text{where } s_k := \begin{cases} +1 & \text{if } z_k = 1, \\ -1 & \text{if } z_k = 0. \end{cases}$$

Thus the number of Alice points is $m_A = nB = n^2$. Alice runs the one-pass algorithm $\mathcal{A}$ on this prefix and sends its memory state (s bits) to Bob.

Bob's suffix. Given index $i \in [n]$, Bob appends

$$T \text{ copies of } (x, y) = (2i - 1,\ -1) \qquad \text{and} \qquad T \text{ copies of } (x, y) = (2i + 1,\ +1).$$

Let the full stream be $S = S_A(z) \circ S_B(i)$, whose total length is

$$m = m_A + 2T = n^2 + 8n^2 = 9n^2.$$

Define two adjacent split positions

$$j^- := 2i - 1, \qquad j^+ := 2i.$$

Note that j^- places the $x = 2i$ block on the right, while j^+ places it on the left.

▷ Claim 17. For every $j \notin \{j^-, j^+\}$, we have $L_{\text{mis}}(j) \geq 4/9$.

Proof. Consider any $j \leq 2i - 2$. Then both anchor blocks $(2i - 1, -1)$ and $(2i + 1, +1)$ lie on the right side $[j + 1, N]$. Hence

$$f_{-1,[j+1,N]} \geq T, \qquad f_{+1,[j+1,N]} \geq T,$$

so $\min\{f_{-1,[j+1,N]}, f_{+1,[j+1,N]}\} \geq T$ and therefore

$$L_{\text{mis}}(j) \;\geq\; \frac{T}{m} \;=\; \frac{4n^2}{9n^2} \;=\; \frac{4}{9}.$$

A symmetric argument applies to any $j \geq 2i+1$: then both anchors lie on the left side $[1, j]$, giving $\min\{f_{-1,[1,j]}, f_{+1,[1,j]}\} \geq T$ and again $L_{\text{mis}}(j) \geq 4/9$. ◁

▷ Claim 18. For each $j \in \{j^-, j^+\}$, we have $L_{\text{mis}}(j) \leq 1/9$. Moreover, the left majority label is -1 and the right majority label is $+1$. Consequently,

$$\text{OPT} = \min\{L_{\text{mis}}(j^-), L_{\text{mis}}(j^+)\}.$$

Proof. Fix $j \in \{j^-, j^+\}$. Then all T copies of $(2i-1, -1)$ lie on the left side, and all T copies of $(2i+1, +1)$ lie on the right side.

Since $T = 4n^2 > n^2 = m_A$, the left side contains more -1 labels from Bob than the total number of Alice points, so the left majority label is -1. Similarly, the right majority label is $+1$.

Thus, under the optimal majority vote on each side, all Bob points are classified correctly. Only Alice's points can be misclassified, and there are exactly $m_A = n^2$ such points. Therefore

$$L_{\text{mis}}(j) \leq \frac{m_A}{m} = \frac{n^2}{9n^2} = \frac{1}{9}.$$

By Claim 17, every $j \notin \{j^-, j^+\}$ satisfies

$$L_{\text{mis}}(j) \geq \frac{4}{9} > \frac{1}{9},$$

so the optimum is attained at one of j^-, j^+. ◁

▷ Claim 19. $|L_{\text{mis}}(j^-) - L_{\text{mis}}(j^+)| > \varepsilon$, and $\arg\min\{L_{\text{mis}}(j^-), L_{\text{mis}}(j^+)\}$ reveals z_i.

Proof. Let G be the block at $x = 2i$, consisting of $B = n$ copies of label $s_i \in \{-1, +1\}$ (where $s_i = +1$ iff $z_i = 1$, and $s_i = -1$ iff $z_i = 0$). All points other than G lie on the same side under both j^- and j^+, and Claim 18 shows the side-majorities remain -1 on the left and $+1$ on the right. Thus the only change in misclassification count comes from moving G from right (under j^-) to left (under j^+).

Case 1: $z_i = 1$. (so $s_i = +1$). Under j^-, G lies on the right (majority $+1$), contributing 0 to $f_{-1,(j^-,N]}$. Under j^+, G lies on the left (majority -1), contributing B to $f_{+1,[1,j^+]}$. Hence $L_{\text{mis}}(j^+) = L_{\text{mis}}(j^-) + B/m$.

Case 2: $z_i = 0$. (so $s_i = -1$). Under j^-, G lies on the right (majority $+1$), contributing B to $f_{-1,(j^-,N]}$. Under j^+, G lies on the left (majority -1), contributing 0 to $f_{+1,[1,j^+]}$. Hence $L_{\text{mis}}(j^-) = L_{\text{mis}}(j^+) + B/m$.

In both cases $|L_{\text{mis}}(j^-) - L_{\text{mis}}(j^+)| = B/m = 1/(9n) \geq 10\varepsilon > \varepsilon$, and the minimizer reveals z_i. ◁

Since $\text{OPT} = \min\{L_{\text{mis}}(j^-), L_{\text{mis}}(j^+)\}$ and $|L_{\text{mis}}(j^-) - L_{\text{mis}}(j^+)| > \varepsilon$, any algorithm outputting $\widehat{j}$ with $L_{\text{mis}}(\widehat{j}) \leq \text{OPT} + \varepsilon$ must output the correct minimizer in $\{j^-, j^+\}$ with probability at least $2/3$, enabling Bob to recover z_i. Thus, the memory state (s bits) yields a one-way protocol for INDEX with communication s, so by Lemma 10, $s = \Omega(n) = \Omega(1/\varepsilon)$. ◀

3.3 Lower bound for classification under Gini impurity

Using the re-encoding trick, we can easily prove a space lower bound for the loss based on the Gini impurity.

▶ **Theorem 20.** *Fix $\varepsilon \in (0, 10^{-3})$. Any one-pass randomized streaming algorithm that, with probability at least $2/3$, outputs a split $\widehat{j} \in [N]$ satisfying*

$$L_{\mathrm{Gini}}(\widehat{j}) \leq \mathrm{OPT} + \varepsilon$$

must use $\Omega(1/\varepsilon)$ bits of memory, even for instances with $N = \Theta(1/\varepsilon)$.

References

1 Farid M. Ablayev. Lower bounds for one-way probabilistic communication complexity and their application to space complexity. *Theor. Comput. Sci.*, 157(2):139–159, 1996. doi:10.1016/0304-3975(95)00157-3.

2 Samaneh Aminikhanghahi and Diane J Cook. A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367, 2017. doi:10.1007/S10115-016-0987-Z.

3 Daniel Nowak Assis, Jean Paul Barddal, and Fabrício Enembreck. Adaptive options for decision trees in evolving data stream classification. In *ECML/PKDD (7)*, Lecture Notes in Computer Science, pages 327–344. Springer, 2025. doi:10.1007/978-3-032-06109-6_19.

4 Albert Bifet, Geoff Holmes, Bernhard Pfahringer, Richard Kirkby, and Ricard Gavalda. New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 139–148, 2009.

5 Albert Bifet, Jiajin Zhang, Wei Fan, Cheng He, Jianfeng Zhang, Jianfeng Qian, Geoff Holmes, and Bernhard Pfahringer. Extremely fast decision tree mining for evolving data streams. In *KDD*, pages 1733–1742. ACM, 2017. doi:10.1145/3097983.3098139.

6 Leo Breiman, J. H. Friedman, Richard A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

7 Alberto Cano and Bartosz Krawczyk. ROSE: robust online self-adjusting ensemble for continual learning on imbalanced drifting data streams. *Mach. Learn.*, 111(7):2561–2599, 2022. doi:10.1007/S10994-022-06168-X.

8 Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *KDD*, pages 785–794. ACM, 2016. doi:10.1145/2939672.2939785.

9 Jie Cheng, Usama M. Fayyad, Keki B. Irani, and Zhaogang Qian. Improved decision trees: A generalized version of ID3. In *ML*, pages 100–106. Morgan Kaufmann, 1988.

10 Graham Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, 55(1):58–75, 2005. doi:10.1016/j.jalgor.2003.12.001.

11 Pedro M. Domingos and Geoff Hulten. Mining high-speed data streams. In *KDD*, pages 71–80. ACM, 2000. doi:10.1145/347090.347107.

12 James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *ICML*, pages 194–202. Morgan Kaufmann, 1995. doi:10.1016/B978-1-55860-377-6.50032-3.

13 Usama M. Fayyad and Keki B. Irani. On the handling of continuous-valued attributes in decision tree generation. *Mach. Learn.*, 8:87–102, 1992. doi:10.1007/BF00994007.

14 Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997. doi:10.1006/jcss.1997.1504.

15 Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

16 Heitor Murilo Gomes, Albert Bifet, Jesse Read, Jean Paul Barddal, Fabrício Enembreck, Bernhard Pfahringer, Geoff Holmes, and Talel Abdessalem. Adaptive random forests for evolving data stream classification. *Mach. Learn.*, 106(9-10):1469–1495, 2017. doi:10.1007/S10994-017-5642-8.

17 Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *NeurIPS*, 2022.

18 Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer, 2009. doi:10.1007/978-0-387-84858-7.

19 Tin Kam Ho. Random decision forests. In *ICDAR*, pages 278–282. IEEE Computer Society, 1995. doi:10.1109/ICDAR.1995.598994.

20 Dug Hun Hong, Sungho Lee, and Kyung Tae Kim. A note on the handling of fuzziness for continuous-valued attributes in decision tree generation. In *FSKD*, volume 4223 of *Lecture Notes in Computer Science*, pages 241–245. Springer, 2006. doi:10.1007/11881599_27.

21 Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106, 2001. doi:10.1145/502512.502529.

22 Laurent Hyafil and Ronald L. Rivest. Constructing optimal binary decision trees is np-complete. *Inf. Process. Lett.*, 5(1):15–17, 1976. doi:10.1016/0020-0190(76)90095-8.

23 Ruoming Jin and Gagan Agrawal. Efficient decision tree construction on streaming data. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 571–576, 2003. doi:10.1145/956750.956821.

24 Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *NIPS*, pages 3146–3154, 2017. URL: https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

25 Chaitanya Manapragada, Geoffrey I Webb, and Mahsa Salehi. Extremely fast decision tree. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1953–1962, 2018. doi:10.1145/3219819.3220005.

26 Llew Mason, Jonathan Baxter, Peter L. Bartlett, and Marcus R. Frean. Boosting algorithms as gradient descent. In *NIPS*, pages 512–518. The MIT Press, 1999. URL: http://papers.nips.cc/paper/1766-boosting-algorithms-as-gradient-descent.

27 Jacob Montiel, Max Halford, Saulo Martiello Mastelini, Geoffrey Bolmier, Raphael Sourty, Robin Vaysse, Adil Zouitine, Heitor Murilo Gomes, Jesse Read, Talel Abdessalem, and Albert Bifet. River: machine learning for streaming data in python. *Journal of Machine Learning Research*, 22(110):1–8, 2021. URL: http://jmlr.org/papers/v22/20-1380.html.

28 Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11:285–296, 1975.

29 Huy Pham, Hoang Ta, and Hoa T. Vu. Constructing decision trees from data streams. In *ISIT*, pages 1–6. IEEE, 2025. doi:10.1109/ISIT63088.2025.11195218.

30 Liudmila Ostroumova Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *NeurIPS*, pages 6639–6649, 2018. URL: https://proceedings.neurips.cc/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html.

31 J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

32 Leszek Rutkowski, Maciej Jaworski, Lena Pietruczuk, and Piotr Duda. The cart decision tree for mining data streams. *Information Sciences*, 266:1–15, 2014. doi:10.1016/J.INS.2013.12.060.

33 Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Inf. Fusion*, 81:84–90, 2022. doi:10.1016/J.INFFUS.2021.11.011.

34 Paula Raissa Silva, João Vinagre, and João Gama. Fed-vfdt: Federated very fast decision trees with coordinated splitting over data streams. In *ICTAI*, pages 653–660. IEEE, 2025. doi:10.1109/ICTAI66417.2025.00094.

35 Nikolaj Tatti. Approximating splits for decision trees quickly in sparse data streams. In *SDM*, pages 647–655. SIAM, 2025. doi:10.1137/1.9781611978520.69.
36 Haixun Wang, Wei Fan, Philip S. Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *KDD*, pages 226–235. ACM, 2003. doi:10.1145/956750.956778.

A Omitted Proofs

Proof of Lemma 5. Let

$$A := \{i : x_i \leq j\}, \qquad B := \{i : j < x_i \leq j'\}, \qquad C := \{i : x_i > j'\}.$$

Then $|B| = b$. Thus split j corresponds to the partition

$$(A,\ B \cup C),$$

while split j' corresponds to

$$(A \cup B,\ C).$$

Let α and β be values (one can show that $\alpha = \mu(j)$ and $\beta = \gamma(j)$) such that

$$L_{\text{MSE}}(j) = \frac{1}{m}\left(\sum_{i \in A}(y_i - \alpha)^2 + \sum_{i \in B \cup C}(y_i - \beta)^2\right).$$

Since $L_{\text{MSE}}(j')$ is the minimum possible loss for split j', we may upper bound it by using the same constants α and β:

$$L_{\text{MSE}}(j') \leq \frac{1}{m}\left(\sum_{i \in A \cup B}(y_i - \alpha)^2 + \sum_{i \in C}(y_i - \beta)^2\right).$$

Subtracting the two expressions gives

$$L_{\text{MSE}}(j') - L_{\text{MSE}}(j) \leq \frac{1}{m}\sum_{i \in B}\Big((y_i - \alpha)^2 - (y_i - \beta)^2\Big).$$

Because $y_i, \alpha, \beta \in [0, M]$, each term lies in $[-M^2, M^2]$, so each summand is at most M^2. Hence

$$L_{\text{MSE}}(j') - L_{\text{MSE}}(j) \leq \frac{bM^2}{m}.$$

Now reverse the roles of j and j'. Let (α', β') be an optimal pair for split j'. Using (α', β') as a feasible choice for split j, the same argument gives

$$L_{\text{MSE}}(j) - L_{\text{MSE}}(j') \leq \frac{bM^2}{m}.$$

Combining the two bounds,

$$|L_{\text{MSE}}(j') - L_{\text{MSE}}(j)| \leq \frac{bM^2}{m}.$$ ◀

Proof of Lemma 6. If $K \geq m$, then the reservoir contains all stream items, so the claim is trivial.

Assume $K < m$. Fix an interval $(a, b] \subseteq [N]$, and let t be the number of stream items whose feature value lies in $(a, b]$. The final reservoir is a uniformly random K-subset of the m stream items. Hence

$$\Pr[\text{the reservoir misses all data points in } (a,b]] = \frac{\binom{m-t}{K}}{\binom{m}{K}} = \prod_{i=0}^{K-1} \frac{m-t-i}{m-i}$$
$$\leq \prod_{i=0}^{K-1} \left(1 - \frac{t}{m}\right) = \left(1 - \frac{t}{m}\right)^K,$$

where each factor uses $\frac{m-t-i}{m-i} = 1 - \frac{t}{m-i} \leq 1 - \frac{t}{m}$ since $m - i \leq m$. If $t > \tau m$, then

$$\Pr[\text{the reservoir misses all data points in } (a,b]] \leq (1-\tau)^K \leq e^{-K\tau}.$$

Since $K = \lceil C \log N/\tau \rceil$, this is at most N^{-C}.

There are at most N^2 intervals of the form $(a, b] \subseteq [N]$. By a union bound, the probability that some interval containing more than τm points is missed by the reservoir is at most

$$N^2 \cdot N^{-C}.$$

Choosing $C \geq 4$ makes this at most $1/N^2$. ◀

Proof of Lemma 11. Write $(y_t - a) = (y_t - \bar{y}) + (\bar{y} - a)$ and expand:

$$\sum_{t=1}^{k} (y_t - a)^2 = \sum_{t=1}^{k} (y_t - \bar{y})^2 + 2(\bar{y} - a) \sum_{t=1}^{k} (y_t - \bar{y}) + k(\bar{y} - a)^2.$$

The cross term vanishes since $\sum_{t=1}^{k} (y_t - \bar{y}) = 0$, giving the identity. The inequality follows from $k(a - \bar{y})^2 \geq 0$. ◀

Proof of Lemma 8. For a fixed split j, let $L := [1, j]$ and $R := [j + 1, N]$. Re-encode each label $y_i \in \{-1, +1\}$ as $z_i = (y_i + 1)/2 \in \{0, 1\}$, and define

$$L_{\text{MSE}}(j) := \frac{1}{m} \left(\sum_{i:\, x_i \leq j} (z_i - \bar{z}_L)^2 + \sum_{i:\, x_i > j} (z_i - \bar{z}_R)^2 \right),$$

where $\bar{z}_L$ and $\bar{z}_R$ are the means of the z_i's on the left and right sides, respectively.

Let

$$n_L := f_{+1,L} + f_{-1,L}, \qquad n_R := f_{+1,R} + f_{-1,R}.$$

If $n_L > 0$, then $\bar{z}_L = f_{+1,L}/n_L$, and

$$\sum_{i:\, x_i \leq j} (z_i - \bar{z}_L)^2 = f_{+1,L}(1 - \bar{z}_L)^2 + f_{-1,L}(0 - \bar{z}_L)^2$$
$$= f_{+1,L} \left(\frac{f_{-1,L}}{n_L}\right)^2 + f_{-1,L} \left(\frac{f_{+1,L}}{n_L}\right)^2$$
$$= \frac{f_{+1,L} f_{-1,L}^2 + f_{-1,L} f_{+1,L}^2}{n_L^2}$$
$$= \frac{f_{+1,L} f_{-1,L} (f_{+1,L} + f_{-1,L})}{n_L^2} = \frac{f_{+1,L} f_{-1,L}}{n_L}.$$

If $n_L = 0$, then the left side is empty and this contribution is 0. Likewise, the right-side contribution equals $f_{+1,R} f_{-1,R}/n_R$ when $n_R > 0$, and 0 when $n_R = 0$.

Hence

$$2L_{\mathrm{MSE}}(j) = \frac{2}{m}\left(\frac{f_{+1,L}f_{-1,L}}{n_L} + \frac{f_{+1,R}f_{-1,R}}{n_R}\right) = L_{\mathrm{Gini}}(j),$$

where empty-side terms are interpreted as 0. Since $z_i \in [0,1]$ (i.e. $M = 1$), Lemma 5 gives $|L_{\mathrm{MSE}}(j') - L_{\mathrm{MSE}}(j)| \le \ell/m$, and therefore

$$\big|L_{\mathrm{Gini}}(j') - L_{\mathrm{Gini}}(j)\big| = 2\big|L_{\mathrm{MSE}}(j') - L_{\mathrm{MSE}}(j)\big| \le \frac{2\ell}{m}. \qquad \blacktriangleleft$$

Proof of Lemma 12. Let $\mu' := \frac{a\mu + Bv}{a+B}$ be the mean of $D \cup G$. Apply Lemma 11 to D,

$$\sum_{y\in D}(y-\mu')^2 = \mathrm{SSE}(D) + a(\mu-\mu')^2.$$

Since every element of G equals v, we have $\sum_{y\in G}(y-\mu')^2 = B(v-\mu')^2$. Summing,

$$\mathrm{SSE}(D\cup G) = \sum_{y\in D\cup G}(y-\mu')^2 = \mathrm{SSE}(D) + a(\mu-\mu')^2 + B(v-\mu')^2.$$

It remains to show $a(\mu-\mu')^2 + B(v-\mu')^2 = \frac{aB}{a+B}(\mu-v)^2$. Substituting $\mu - \mu' = \frac{B(\mu-v)}{a+B}$ and $v - \mu' = \frac{a(v-\mu)}{a+B}$,

$$a(\mu-\mu')^2 + B(v-\mu')^2 = \frac{aB^2(\mu-v)^2}{(a+B)^2} + \frac{Ba^2(\mu-v)^2}{(a+B)^2} = \frac{aB(a+B)(\mu-v)^2}{(a+B)^2}$$
$$= \frac{aB}{a+B}(\mu-v)^2. \qquad \blacktriangleleft$$

Proof of Theorem 20. Re-encode each label $y_i \in \{-1,+1\}$ as $z_i = (y_i+1)/2 \in \{0,1\}$. By the identity established in the proof of Lemma 8, we have

$$L_{\mathrm{Gini}}(j) = 2\,L_{\mathrm{MSE}}(j)$$

for every split j, where L_{MSE} is the squared loss on the re-encoded stream with $M = 1$. Hence any one-pass algorithm that outputs $\widehat{j}$ with $L_{\mathrm{Gini}}(\widehat{j}) \le \mathrm{OPT} + \varepsilon$ also satisfies $L_{\mathrm{MSE}}(\widehat{j}) \le \mathrm{OPT}_{\mathrm{MSE}} + \varepsilon/2$, giving a one-pass $(\varepsilon/2)$-additive algorithm for regression with $M = 1$. By Theorem 13 with $M = 1$, such an algorithm requires $\Omega(M^2/(\varepsilon/2)) = \Omega(1/\varepsilon)$ bits of memory. $\blacktriangleleft$