Logarithm-depth Streaming Multi-label Decision Trees (Supplementary material)

Abstract

This Supplement presents additional details in support of the full article. These include the proofs of the theoretical statements from the main body of the paper and additional theoretical results. We also provide additional algorithm's pseudo-codes. The Supplement also contains the description of the experimental setup, and additional experiments and figures to provide further empirical support for the proposed methodology.

7 ADDITIONAL THEORETICAL RESULTS

Next lemma shows that in isolation, when the purity of the split is perfect, decreasing the value of the objective leads to recovering more balanced splits.

Lemma 6. *If a node split is perfectly pure, then*

$$\beta \leq J - J^*. \tag{6}$$

Next lemma shows that in isolation, when the balancedness of the split is perfect, decreasing the value of the objective leads to recovering more pure splits.

Lemma 7. *If a node split is perfectly balanced and assuming that the following condition holds: $\lambda_1(M-1) \geq \lambda_2 \geq \lambda_1 \frac{M-1}{2}$, then*

$$\alpha \leq (J + \lambda_2)\frac{2}{M(2\lambda_2 - \lambda_1(M-1))}. \tag{7}$$

Below we provide a new assumption and corresponding theorem that generalizes Theorem 2, by removing the balancedness assumption.

Assumption 7.1. *γ-Weak Hypothesis Assumption: for any distribution $\mathcal{P}$ over the data, at each node of the tree $\mathcal{T}$ there exist a partition such that $\sum_i \pi_i \left|\frac{P_R^i}{P_R} - \frac{P_L^i}{P_L}\right| \geq \gamma$, where $\gamma \in (0, 1]$.*

Theorem 3. *Under the Weak Hypothesis Assumptions 7.1 and 3.2 for any $\alpha \in [0, 1]$ to obtain $e_r(\mathcal{T}) \leq \alpha$ it suffices to have a tree with t internal nodes that satisfy $(t+1) \geq (\frac{1}{\alpha})^{\frac{16 \ln K}{cr^2\gamma^2(1-b)\log_2(e)}}$, where $b = |P_R + P_L - 1|$.*

Below we consider the weak hypothesis assumption that generalizes the Assumption 3.1 to the M-ary case and prove corresponding lemma that generalizes Lemma 5.

Assumption 7.2 (Generalization of Assumption 3.1)**.** *γ-Weak Hypothesis Assumption: for any distribution $\mathcal{P}$ over the data, at each node n of the tree $\mathcal{T}$ there exist a partition such that $\sum_{i=1}^{K}\sum_{j=1}^{M}\sum_{l=1}^{M}\pi_i \left|P_j^i - P_l^i\right| \geq \gamma$, where $\gamma \in (0, 1]$.*

Lemma 8 (Generalization of Lemma 5)**.** *Under the Weak Hypothesis Assumption 7.2, the $e_r(\mathcal{T})$ is monotonically decreasing with every split of the tree.*

7.1 Relation of the Objective to Shannon Entropy and Error Bound (Binary Tree Case)

In this section we first show the relation of the objective J to a classical decision-tree criterion, Shannon entropy, and specifically we demonstrate that minimizing the objective leads to the reduction of this criterion. We restrict ourselves to the case of binary tree. We omit the analysis for the M-ary to avoid over-complicating the notation. The entropy of tree leaves in the case when examples can be sent to multiple directions can be calculated as:

$$G = \sum_{\tilde{\mathcal{L}} \subset \mathcal{L}} w_{\tilde{\mathcal{L}}} \sum_{i=1}^{K} \rho_i^{\tilde{\mathcal{L}}} \ln\left(\frac{1}{\rho_i^{\tilde{\mathcal{L}}}}\right) \tag{8}$$

where $\mathcal{L}$ is the set of all tree leaves, $\tilde{\mathcal{L}}$ is a subset of the leaves (the summation is taken over all the possible subsets), $\rho_i^{\tilde{\mathcal{L}}}$ is the probability that example with label i reaches all the leaves in $\tilde{\mathcal{L}}$, and $w_{\tilde{\mathcal{L}}}$ is the weight of subset of leaves. This weight is defined as the probability that a randomly chosen point from distribution $\mathcal{P}$ reaches all leaves in $\tilde{\mathcal{L}}$. Also note that $\sum_{\tilde{\mathcal{L}}\subset\mathcal{L}} w_{\tilde{\mathcal{L}}} = 1$ and $w_{\tilde{\mathcal{L}}=\varnothing} = 0$.

Theorem 4. *Under the Weak Hypothesis Assumptions 3.1 and 3.2, and an additional assumption that each node produces perfectly balanced split, for any $\kappa \in [0, \ln K]$ to obtain $G_t^e \leq \kappa$ it suffices to have a tree with t internal nodes that satisfy*

$$(t+1) \geq (\frac{G_1}{\kappa})^{\frac{16 \ln K}{cr^2\gamma^2(1-b)\log_2(e)}},$$

where $b = |P_R + P_L - 1|$.

8 THEORETICAL PROOFS

Proof of Lemma 1. We rewrite the objective using the total law of probability:

$$J = \left|\sum_{i=1}^{K} \pi_i(P_R^i - P_L^i)\right| - \lambda_1 \sum_{i=1}^{K} \pi_i \left|P_R^i - P_L^i\right| + \lambda_2 \left|\sum_{i=1}^{K} \pi_i(P_R^i + P_L^i) - 1\right|, \tag{9}$$

where $P_R^i, P_L^i \in [0,1]$ for all $i = 1, 2, \ldots, K$. The objective admits optimum on the extremes of the $[0,1]$ interval. Therefore, we define the following:

$$L_1 = \{i : i \in \{1,\ldots,K\}, P_R^i = 1 \;\&\; P_L^i = 1\}, \quad L_2 = \{i : i \in \{1,\ldots,K\}, P_R^i = 0 \;\&\; P_L^i = 0\}, \tag{10}$$

$$L_3 = \{i : i \in \{1,\ldots,K\}, P_R^i = 1 \;\&\; P_L^i = 0\}, \quad L_4 = \{i : i \in \{1,\ldots,K\}, P_R^i = 0 \;\&\; P_L^i = 1\} \tag{11}$$

By substituting the above in the objective we have:

$$J = \left|\sum_{i\in L_3} \pi_i - \sum_{i\in L_4} \pi_i\right| - \lambda_1 \sum_{i\in(L_3\cup L_4)} \pi_i + \lambda_2 \left|\sum_{i\in(L_3\cup L_4)} \pi_i + \sum_{i\in L_1} 2\pi_i - 1\right|. \tag{12}$$

We send each example either to the right, left or both directions:

$$\sum_{i\in(L_1\cup L_3\cup L_4)} \pi_i = \sum_{i\in L_1} \pi_i + \sum_{i\in L_3} \pi_i + \sum_{i\in L_4} \pi_i = 1. \tag{13}$$

Thus we can further write

$$J = \left|1 - \sum_{i\in L_1} \pi_i - 2\sum_{i\in L_4} \pi_i\right| - \lambda_1(1 - \sum_{i\in L_1} \pi_i) + \lambda_2 \sum_{i\in L_1} \pi_i. \tag{14}$$

For ease of notation, we define $a \coloneqq \sum_{i\in L_4} \pi_i$, $a' \coloneqq \sum_{i\in L_3} \pi_i$, and $b \coloneqq \sum_{i\in L_1} \pi_i$. Therefore

$$J = |1 - b - 2a| - \lambda_1(1-b) + \lambda_2 b = |b + 2a' - 1| - \lambda_1(1-b) + \lambda_2 b, \tag{15}$$

where $a, b \in [0,1]$. Since we are interested in bounding J, we consider the values of a and b at the extremes of $[0,1]$ interval:

$$\text{if } a = 1 \text{ then } b = 0 \;\rightarrow\; J = 1 - \lambda_1, \qquad \text{if } b = 1 \text{ then } a = 0 \;\rightarrow\; J = \lambda_2 \tag{16}$$

$$\text{if } a = 0 \text{ then} \begin{cases} b = 0 \; (a' = 1) & \rightarrow \quad J = 1 - \lambda_1 \\ b = 1 & \rightarrow \quad J = \lambda_2 \end{cases} \tag{17}$$

$$\text{if } b = 0 \text{ then} \begin{cases} a = 0 \; (a' = 1) & \rightarrow \quad J = 1 - \lambda_1 \\ a = 1 & \rightarrow \quad J = 1 - \lambda_1 \\ a = 0.5 & \rightarrow \quad J = -\lambda_1 \end{cases} \tag{18}$$

Therefore $J \in [-\lambda_1,\ \lambda_2]$.
Next, we show that the perfectly balanced and pure split is attained at the minimum of the objective. The perfectly balanced split is achieved when $P_R = P_L$ and then the balancing term in the objective becomes zero. The perfectly pure split is achieved when the class integrity term in the objective satisfies $\sum_{i=1}^{K} \pi_i \left|P_R^i - P_L^i\right| = \sum_{i=1}^{K} \pi_i = 1$. Simultaneously, the following holds $\sum_{i=1}^{K} \pi_i (P_R^i + P_L^i) = 1$, and therefore the multi-way penalty is zero as well. Thus, $J = 0 - \lambda_1 + 0 = -\lambda_1$. In order to prove the opposite direction of the claim, recall that the minimum of the objective occurs for $b = 0$ and $a = 0.5$. Since $a + a' + b = 1$, therefore $a' = 0.5$. This corresponds to the perfectly pure and balanced split. ☐

Proof of Lemma 2. $P_j^i \in [0,\ 1]$ for all $i = 1, 2, \ldots, K$ and $j = 1, 2, \ldots, M$. The objective admits optimum on the extremes of the $[0, 1]$ interval. In the following proof we consider a different approach than in the proof of Lemma 1. In order to get the minimum of the objective, we try to minimize each of its terms separately and on the top of that incorporate their correlations. For now, we assume that the first term, the balancing term, is minimized and therefore is equal to zero. We define case C_n as the scenario when for any $i = 1, 2, \ldots, K$, $P_j^i = 1$ for n "directions" ($n \leq M$), i.e. n distinct js such that $j \in \{1,\ 2, \ldots,\ M\}$, and $P_j^i = 0$ for the remaining j's. The class integrity and multi-way penalty terms can then be derived as follows:

$$J_{\text{class integrity term}|C_n} = \lambda_1 \sum_{i=1}^{K} \sum_{j=1}^{M} \sum_{l=j+1}^{M} \pi_i \left|P_j^i - P_l^i\right| = n(M-n), \tag{19}$$

$$J_{\text{multi-way penalty term}|C_n} = \lambda_2 \left(\sum_{j=1}^{M} P_j \right) - 1 = n - 1. \tag{20}$$

Therefore, the objective value would then become: $J = -\lambda_1 n(M-n) + \lambda_2(n-1)$. We aim to have the minimum of the objective for perfectly pure split. The perfectly pure split is achieved when case C_1 holds. Therefore, we need:

$$-\lambda_1(M-1) < -\lambda_1 n(M-n) + \lambda_2(n-1) \quad \text{for } n \in \{2, \ldots, M\}. \tag{21}$$

The lower-bound of the right side is achieved for $n = 2$:

$$-\lambda_1(M-1) < -\lambda_1 2(M-2) + \lambda_2 \quad \rightarrow \quad M - 3 < \frac{\lambda_2}{\lambda_1}. \tag{22}$$

With the above condition, the minimum of the objective is equal to $-\lambda_1(M-1)$. Note that our first assumption on the balancing term can still hold for all C_n cases. Therefore, we have shown that the minimum of the objective corresponds to the perfectly pure and balanced split.
In order to get the upper-bound for J, we first show that $J_{\text{balancing term}} \leq J_{\text{class integrity term}}$ as follows:

$$J_{\text{balancing term}} = \sum_{j=1}^{M} \sum_{l=j+1}^{M} |P_j - P_l| = \sum_{j=1}^{M} \sum_{l=j+1}^{M} \left| \sum_{i=1}^{K} \pi_i (P_j^i - P_l^i) \right| \tag{23}$$

$$\leq \sum_{j=1}^{M} \sum_{l=j+1}^{M} \sum_{i=1}^{K} \pi_i \left|P_j^i - P_l^i\right| = J_{\text{class integrity term}}. \tag{24}$$

Therefore, the maximum of the summation of the terms is achieved when $J_{\text{balancing term}} = J_{\text{class integrity term}}$. The maximum of the multi-way penalty term is attained when sending all examples to every direction, resulting in $J_{\text{multi-way penalty term}} = (M-1)$. In this case, $J_{\text{balancing term}} = J_{\text{class integrity term}} = 0$, and thus, $J = \lambda_2(M-1)$. Hence, we have $J \in [-\lambda_1(M-1),\ \lambda_2(M-1)]$. ☐

Proof of Lemma 6. The perfectly pure split is attained when $P_j^i = 1$ for only one value of j, and $P_j^i = 0$ for the remaining j's. This leads the class integrity term to satisfy $\sum_{j=1}^{M} \sum_{l=j+1}^{M} \sum_{i=1}^{K} \pi_i \left|P_j^i - P_l^i\right| = (M-1)$ and the multi-way penalty term to satisfy $\sum_{i=1}^{k} \pi_i \sum_{j=1}^{M} P_j^i - 1 = 0$. Thus we have:

$$J - J^* = \sum_{j=1}^{M} \sum_{l=j+1}^{M} |P_j - P_l| \tag{25}$$

$$= \sum_{j=1}^{M} \sum_{l=j+1}^{M} \left| \left(P_j - \frac{\sum_{i=1}^{M} P_i}{M} \right) - \left(P_l - \frac{\sum_{i=1}^{M} P_i}{M} \right) \right|. \tag{26}$$

Let $j^* = \text{argmax}_{j\in\{1,2,\ldots,M\}}|P_j - \frac{\sum_{i=1}^M P_i}{M}|$. Without loss of generality assume $P_{j^*} - \frac{\sum_{i=1}^M P_i}{M} \geq 0$ and in that case there exists an l^* such that $P_{l^*} - \frac{\sum_{i=1}^M P_i}{M} \leq 0$. Therefore we have:

$$J - J^* \geq \left|\left(P_{j^*} - \frac{\sum_{i=1}^M P_i}{M}\right) - \left(P_{l^*} - \frac{\sum_{i=1}^M P_i}{M}\right)\right| \tag{27}$$

$$\geq \left|(P_{j^*} - \frac{\sum_{i=1}^M p_i}{M})\right| = \beta. \tag{28}$$

□

Proof of Lemma 3. Consider a split with a fixed purity factor α. J^α_{purity} denotes the sum of the class integrity and multi-way penalty terms of the objective function. When subtracting them from the total value of the objective at node n we obtain the balancing term. Thus we have:

$$J - J^\alpha_{\text{purity}} = \sum_{j=1}^M \sum_{l=j+1}^M |P_j - P_l| \tag{29}$$

$$= \sum_{j=1}^M \sum_{l=j+1}^M \left|\left(P_j - \frac{\sum_{i=1}^M P_i}{M}\right) - \left(P_l - \frac{\sum_{i=1}^M P_i}{M}\right)\right|. \tag{30}$$

Let $j^* = \text{argmax}_{j\in\{1,2,\ldots,M\}}|P_j - \frac{\sum_{i=1}^M P_i}{M}|$. Without loss of generality assume $P_{j^*} - \frac{\sum_{i=1}^M P_i}{M} \geq 0$ and in that case there exists an l^* such that $P_{l^*} - \frac{\sum_{i=1}^M P_i}{M} \leq 0$. Therefore we have:

$$J - J^\alpha_{\text{purity}} \geq \left|\left(P_{j^*} - \frac{\sum_{i=1}^M P_i}{M}\right) - \left(P_{l^*} - \frac{\sum_{i=1}^M P_i}{M}\right)\right| \tag{31}$$

$$\geq \left|(P_{j^*} - \frac{\sum_{i=1}^M p_i}{M})\right| = \beta. \tag{32}$$

□

Proof of Lemma 7. The perfectly balanced split is attained when $P_1 = P_2 = ... = P_M$. This zeros out the balancing term in the objective function. Hence:

$$J = -\lambda_1 \sum_{i=1}^K \sum_{j=1}^M \sum_{l=j+1}^M \pi_i \left|P^i_j - P^i_l\right| + \lambda_2 \left(\sum_{j=1}^M P_j - 1\right) \tag{33}$$

$$= -\lambda_1 \sum_{i=1}^K \sum_{j=1}^M \sum_{l=j+1}^M \pi_i \left|P^i_j - P^i_l\right| + \lambda_2 \left(\sum_{i=1}^K \sum_{j=1}^M \pi_i P^i_j - 1\right) \tag{34}$$

$$\geq -\lambda_1 \frac{M-1}{2} \sum_{i=1}^K \sum_{j=1}^M \pi_i P^i_j + \lambda_2 \left(\sum_{i=1}^K \sum_{j=1}^M \pi_i P^i_j - 1\right). \tag{35}$$

Thus we have:

$$J + \lambda_2 \geq \left(\lambda_2 - \lambda_1 \frac{M-1}{2}\right) \sum_{i=1}^K \sum_{j=1}^M \pi_i P^i_j \tag{36}$$

$$\geq \left(\lambda_2 - \lambda_1 \frac{M-1}{2}\right) \sum_{i=1}^K \sum_{j=1}^M \pi_i \min(P^i_j, \sum_{l=1}^M P^i_l - P^i_j) \tag{37}$$

$$\geq \left(\lambda_2 - \lambda_1 \frac{M-1}{2}\right) M\alpha. \tag{38}$$

□

Proof of Lemma 4. Consider a split with a fixed balancedness factor β. $J^{\beta}_{\text{balance}}$ denotes the balancing term of the objective function. When subtracting it from the total value of the objective at node n we will obtain the sum of the class integrity and multi-way penalty terms. Hence:

$$J - J^{\beta}_{\text{balance}} = -\lambda_1 \sum_{i=1}^{K} \sum_{j=1}^{M} \sum_{l=j+1}^{M} \pi_i \left| P^i_j - P^i_l \right| + \lambda_2 \left(\sum_{j=1}^{M} P_j - 1 \right) \tag{39}$$

$$= -\lambda_1 \sum_{i=1}^{K} \sum_{j=1}^{M} \sum_{l=j+1}^{M} \pi_i \left| P^i_j - P^i_l \right| + \lambda_2 \left(\sum_{i=1}^{K} \sum_{j=1}^{M} \pi_i P^i_j - 1 \right) \tag{40}$$

$$\geq -\lambda_1 \frac{M-1}{2} \sum_{i=1}^{K} \sum_{j=1}^{M} \pi_i P^i_j + \lambda_2 \left(\sum_{i=1}^{K} \sum_{j=1}^{M} \pi_i P^i_j - 1 \right). \tag{41}$$

Thus we have:

$$J - J^{\beta}_{\text{balance}} + \lambda_2 \geq \left(\lambda_2 - \lambda_1 \frac{M-1}{2} \right) \sum_{i=1}^{K} \sum_{j=1}^{M} \pi_i P^i_j \tag{42}$$

$$\geq \left(\lambda_2 - \lambda_1 \frac{M-1}{2} \right) \sum_{i=1}^{K} \sum_{j=1}^{M} \pi_i \min(P^i_j, \sum_{l=1}^{M} P^i_l - P^i_j) \tag{43}$$

$$\geq \left(\lambda_2 - \lambda_1 \frac{M-1}{2} \right) M\alpha. \tag{44}$$

□

Proof of Theorem 4. In our algorithm, we recursively find the leaf node with the heaviest weight and decide to partition it to two children. Suppose, after t splits the leaf node n has the highest weight, namely w_n, which will be denoted with w for brevity. This weight is defined as the probability that a randomly chosen data point x drawn from a fixed distribution $\mathcal{P}$ reaches the leaf. Let $w_{R\text{ only}}$ and $w_{L\text{ only}}$ be the weight of examples reaching only to the right and left child of node n, and w_{both} be the weight of examples reaching to both children. Also let $P_{both} = |P_R + P_L - 1|$. Note that $w_{R\text{ only}} = wP_{R\text{ only}} = w(P_R - P_{both})$ and $w_{L\text{ only}} = wP_{L\text{ only}} = w(P_L - P_{both})$. Let $\boldsymbol{\rho}$ be a vector with K elements, which its i^{th} element is ρ_i. Furthermore, let $\boldsymbol{\rho}_R$, and $\boldsymbol{\rho}_L$ be K-element vectors with $\rho_{i,R}$ and $\rho_{i,L}$ at its i^{th} entry. Note that $\rho_{i,R} = \frac{\rho_i P^i_R}{P_R}$, and $\rho_{i,L} = \frac{\rho_i P^i_L}{P_L}$. Before the node partition the contribution of node n to the total entropy-based objective is $w\tilde{G}(\boldsymbol{\rho})$. After the split this contribution will be $w_{R\text{ only}}\tilde{G}(\boldsymbol{\rho}_R) + w_{L\text{ only}}\tilde{G}(\boldsymbol{\rho}_L) + w_{both}\tilde{G}(\boldsymbol{\rho})$ (Note that for the examples being sent to both directions we average the histograms of the left and right children. Also note that $(w_{R\text{ only}} + w_{R\text{ only}} + w_{both}) = 1$) Therefore, we have:

$$\Delta_t := G_t - G_{t+1} = w[\tilde{G}(\boldsymbol{\rho}) - P_{R\text{ only}}\tilde{G}(\boldsymbol{\rho}_R) - P_{L\text{ only}}\tilde{G}(\boldsymbol{\rho}_L) - P_{\text{both}}\tilde{G}(\boldsymbol{\rho})] \tag{45}$$

$$= w[\tilde{G}(\boldsymbol{\rho}) - (P_R - P_{both})\tilde{G}(\boldsymbol{\rho}_R) - (P_L - P_{both})\tilde{G}(\boldsymbol{\rho}_L) - P_{both}\tilde{G}(\boldsymbol{\rho})]. \tag{46}$$

Recall that the Shannon entropy is strongly concave with respect to l_1-norm (see Shalev-Shwartz, 2012, Example 2.5), and $\boldsymbol{\rho} = (P_R - \frac{1}{2}P_{both})\boldsymbol{\rho}_R + (P_L - \frac{1}{2}P_{both})\boldsymbol{\rho}_L$, where $P_{both} = P_R + P_L - 1$. Without loss of generality assume $P_R = P_L + \eta$. Hence we re-write Δ_t as follows:

$$\Delta_t = w[(1 - P_{both})\tilde{G}(\boldsymbol{\rho}) - (\frac{1 + \eta - P_{both}}{2})\tilde{G}(\boldsymbol{\rho}_R) - (\frac{1 - \eta - P_{both}}{2})\tilde{G}(\boldsymbol{\rho}_L)] \tag{47}$$

$$= w(1 - P_{both})[\tilde{G}(\boldsymbol{\rho}) - (\frac{1 + \eta - P_{both}}{2(1 - P_{both})})\tilde{G}(\boldsymbol{\rho}_R) - (\frac{1 - \eta - P_{both}}{2(1 - P_{both})})\tilde{G}(\boldsymbol{\rho}_L)]. \tag{48}$$

(49)

We can then use the result from Theorem 2.1.9 in Nestrov (2004):

$$\Delta_t \geq w(1-P_{both})\left[\frac{1}{8}||\boldsymbol{\rho}_R - \boldsymbol{\rho}_L||_1^2\right] \tag{50}$$

$$= w(1-P_{both})r^2\left[\frac{1}{8}||\boldsymbol{\pi}_R - \boldsymbol{\pi}_L||_1^2\right] \tag{51}$$

$$= w(1-P_{both})r^2\left[\frac{1}{8}\left(\sum_{i=1}^{K}\left|\frac{\pi_i P_R^i}{P_R} - \frac{\rho_i P_L^i}{P_L}\right|\right)^2\right]. \tag{52}$$

Here we use the assumption that we have a balance split, i.e. $P_R = P_L$, therefore we continue as follows:

$$= w(1-P_{both})\frac{r^2}{8P_R^2}\left(\sum_{i=1}^{K}\pi_i|P_R^i - P_L^i|\right)^2 \tag{53}$$

$$\geq w(1-P_{both})\frac{r^2}{8}\left(\sum_{i=1}^{K}\pi_i|P_R^i - P_L^i|\right)^2. \tag{54}$$

Now by applying the WHA 3.2:

$$\Delta_t \geq w(1-b)\frac{r^2}{8}\gamma^2. \tag{55}$$

Note that by WHA 3.2 $b \in [0, 1)$. Also note that $w \geq \frac{G_t c}{(t+1)\ln K}$. This comes from the fact that at each step we choose the leaf node with maximum weight. Hence with WHA2, $w = \max_{l\in\mathcal{L}} w_l \geq \frac{c}{(t+1)}$. Also note that uniform distribution maximizes the entropy, i.e. $G_t \leq \ln K$. Accordingly we have:

$$\Delta_t \geq \frac{G_t c}{(t+1)\ln K}[\frac{r^2}{8}\gamma^2(1-b)]. \tag{56}$$

By letting $\eta = \frac{1}{2}\sqrt{\frac{cr^2\gamma^2(1-b)}{2\ln K}}$, we have $\Delta_t \geq \frac{\eta^2 G_t}{(t+1)}$. Thus, we have the following recursion inequality:

$$G_{t+1} \leq G_t - \Delta_t \leq G_t - \frac{\eta^2 G_t}{(t+1)} = G_t[1 - \frac{\eta^2}{(t+1)}]. \tag{57}$$

Then by applying the same proof technique as in Kearns and Mansour (1999) we get the following relationship:

$$G_{t+1} \leq G_1 e^{-\eta^2 \log_2(t+1)/2}. \tag{58}$$

Therefore, to reduce $G_{t+1} \leq \kappa$ it suffices to have (t+1) splits such that $\log_2(t+1) \geq \ln(\frac{G_1}{\kappa})^{\frac{2}{\eta^2}}$. Substituting $\log_2(t+1) = \ln(t+1)\log_2(e)$ results in:

$$\ln(t+1) \geq \ln(\frac{G_1}{\kappa})^{\frac{2}{\eta^2\log_2(e)}} \Leftrightarrow (t+1) \geq (\frac{G_1}{\kappa})^{\frac{2}{\eta^2\log_2(e)}}. \tag{59}$$

□

We next proceed to the proof of Theorem 2.

Proof of Theorem 2. This proof follows the proof of the Theorem 4. Below we directly calculate the error bound. Recall $w_{\tilde{\mathcal{L}}}$ to be the probability that a data point x reached the subset of leaves $\tilde{\mathcal{L}}$. Recall that $\rho_i^{\tilde{\mathcal{L}}}$ is the probability that the data point x has label i given that x reached $\tilde{\mathcal{L}}$, i.e. $\rho_i^{\tilde{\mathcal{L}}} = P(i \in t(x)|x \text{ reached } \tilde{\mathcal{L}})$. Note that each example has r labels, and let's assume we assign first majority r labels from the $\rho_i^{\tilde{\mathcal{L}}}$ histogram to any example reaching $\tilde{\mathcal{L}}$, i.e. $y_r(x) = \{j_1, j_2, ..., j_r\}$, where $j_1 = \text{argmax}_{k\in\{1,2,\dots,K\}}(\rho_k^{\tilde{\mathcal{L}}})$, $j_2 = \text{argmax}_{k\in\{1,2,\dots,K\}\setminus j_1}(\rho_k^{\tilde{\mathcal{L}}})$,..., $j_r = \text{argmax}_{k\in\{1,2,\dots,K\}\setminus\{j_1,\dots,j_{r-1}\}}(\rho_k^{\tilde{\mathcal{L}}})$. We then expand the r-level multi-label error as follows:

$$\epsilon_r(\mathcal{T}) = \frac{1}{r}\sum_{i=1}^{K} P(i\in y_r(x), i\notin t(x)) \tag{60}$$

$$= \frac{1}{r}\sum_{i=1}^{K} P(i\in t(x), i\notin y_r(x)) \tag{61}$$

$$= \frac{1}{r}\sum_{\tilde{\mathcal{L}}\in\mathcal{L}} w_{\tilde{\mathcal{L}}}\sum_{i=1}^{K} P(i\in t(x), i\notin y_r(x)|x \text{ reached } \tilde{\mathcal{L}}) \tag{62}$$

$$= \frac{1}{r}\sum_{\tilde{\mathcal{L}}\in\mathcal{L}} w_{\tilde{\mathcal{L}}}\sum_{\substack{i=1\\ i\neq j_1,\dots,j_R}}^{K} P(i\in t(x)|x \text{ reached } \tilde{\mathcal{L}}) \tag{63}$$

$$= \frac{1}{r}\sum_{\tilde{\mathcal{L}}\in\mathcal{L}} w_{\tilde{\mathcal{L}}}\left(\sum_{i=1}^{K}\rho_i^{\tilde{\mathcal{L}}} - \max_{k\in\{1,2,\dots,K\}}\rho_k^{\tilde{\mathcal{L}}} - \max_{k\in\{1,2,\dots,K\}\setminus j_1}\rho_k^{\tilde{\mathcal{L}}} - \max_{k\in\{1,2,\dots,K\}\setminus\{j_1,j_2\}}\rho_k^{\tilde{\mathcal{L}}} - \cdots - \max_{k\in\{1,2,\dots,K\}\setminus\{j_1,j_2,\dots,j_{r-1}\}}\rho_k^{\tilde{\mathcal{L}}}\right), \tag{64}$$

where $w_{\tilde{\mathcal{L}}}$ denote the probability that example x reaches $\tilde{\mathcal{L}}$ and $\mathcal{L}$ denote the set of all leaves of the tree.

Next we will find the Shannon entropy bound with respect to the error and show that the entropy of the tree, denoted as $G(\mathcal{T})$, upper-bounds the error. Note that:

$$G(\mathcal{T}) = \sum_{\tilde{\mathcal{L}}\in\mathcal{L}} w_{\tilde{\mathcal{L}}}\sum_{i=1}^{K}\rho_i^{\tilde{\mathcal{L}}}\ln\left(\frac{1}{\rho_i^{\tilde{\mathcal{L}}}}\right) \quad\geq\quad \sum_{l\in\mathcal{L}} w_{\tilde{\mathcal{L}}}\sum_{\substack{i=1\\ i\neq j_1,\dots,j_r}}^{K}\rho_i^{\tilde{\mathcal{L}}}\ln\left(\frac{1}{\rho_i^{\tilde{\mathcal{L}}}}\right). \tag{65}$$

Note that $\sum_{i=1}^{K}\rho_i^{\tilde{\mathcal{L}}} = r$. Thus for any $i = 1,2,\dots,K$ such that $i\neq j_1,\dots,j_r$ it must hold that $\rho_i^{\tilde{\mathcal{L}}} \leq \frac{1}{2}$. We continue as follows

$$G(\mathcal{T}) \geq \sum_{\tilde{\mathcal{L}}\in\mathcal{L}} w_{\tilde{\mathcal{L}}}\sum_{\substack{i=1\\ i\neq j_1,\dots,j_r}}^{K}\rho_i^{\tilde{\mathcal{L}}}\ln(2) \tag{66}$$

$$\geq \ln(2)\sum_{\tilde{\mathcal{L}}\in\mathcal{L}} w_{\tilde{\mathcal{L}}}\left(\sum_{i=1}^{K}\rho_i^{\tilde{\mathcal{L}}} - \max_{k\in\{1,2,\dots,K\}}\rho_k^{\tilde{\mathcal{L}}} - \max_{k\in\{1,2,\dots,K\}\setminus j_1}\rho_k^{\tilde{\mathcal{L}}} - \max_{k\in\{1,2,\dots,K\}\setminus\{j_1,j_2\}}\rho_k^{l} - \cdots - \max_{k\in\{1,2,\dots,K\}\setminus\{j_1,j_2,\dots,j_{r-1}\}}\rho_k^{\tilde{\mathcal{L}}}\right)$$

$$= \ln(2) r\epsilon_r(\mathcal{T}) \geq \epsilon_r(\mathcal{T}), \tag{67}$$

where the last inequality comes from the fact that $r\geq 1/\ln(2)$. Now recall that $G_1\leq \ln K$ and normalizing κ in Theorem 4 finishes the proof. □

Proof of Theorem 3. The proof follows the same steps as Theorem 4 until Equation 52. Applying WHA 7.1 at this point will result in the same result as in Equation 55. The rest of the proof would be the same as Theorems 4 and 2. □

Proof of Theorem 1. Since we assume the objective is minimized in every node of the tree, therefore each node is sending examples to only one of its children and consequently each example descends to only one leaf. Thus in any leaf l, we store label histograms and assign first r labels from the histogram to any example reaching that leaf, i.e. $y(x) = \{j_1, j_2, \dots, j_r\}$, where $j_1 = \text{argmax}_{k\in\{1,2,\dots,K\}}\rho_k^l$, $j_2 = \text{argmax}_{k\in\{1,2,\dots,K\}\setminus j_1}(\rho_k^l)$,..., $j_r = \text{argmax}_{k\in\{1,2,\dots,K\}\setminus\{j_1,\dots,j_{r-1}\}}(\rho_k^l)$ and ρ_i^l is the probability that the data point x has label i given that x has reached leaf l, i.e. $\rho_i^l = P(i\in t(x)|x \text{ reached } l)$.

We next expand the r-level multi-label error as follows:

$$\begin{aligned}
\epsilon_r(\mathcal{T}) &= \frac{1}{r}\sum_{i=1}^{K} P(i \in y_r(x), i \notin t(x)) && (68)\\
&= \frac{1}{r}\sum_{i=1}^{K} P(i \in t(x), i \notin y_r(x)) && (69)\\
&= \frac{1}{r}\sum_{l\in\mathcal{L}} w(l)\sum_{i=1}^{K} P(i \in t(x), i \notin y_r(x)|x \text{ reached } l) && (70)\\
&= \frac{1}{r}\sum_{l\in\mathcal{L}} w(l) \sum_{\substack{i=1\\ i\neq j_1,\dots,j_r}}^{K} P(i \in t(x)|x \text{ reached } l) && (71)\\
&= \frac{1}{r}\sum_{l\in\mathcal{L}} w(l)\left(\sum_{i=1}^{K}\rho_i^{(l)} - \max_{k\in\{1,2,\dots,K\}}\rho_k^l - \max_{k\in\{1,2,\dots,K\}\backslash j_1}\rho_k^l \right. && (72)\\
&\qquad \left. - \max_{k\in\{1,2,\dots,K\}\backslash\{j_1,j_2\}}\rho_k^l - \dots - \max_{k\in\{1,2,\dots,K\}\backslash\{j_1,j_2,\dots,j_{r-1}\}}\rho_k^l\right),
\end{aligned}$$

where $w(l)$ denote the probability that example x reaches leaf l and $\mathcal{L}$ denote the set of all leaves of the tree.

From Lemma 1 (for binary tree) and Lemma 2 (for M-ary tree) it follows that for any node in the tree, the corresponding split is balanced and the following holds: $|P_j^i - P_{j'}^i| = 1$ for all labels $i = 1, 2, \dots, K$ and all pairs of children nodes $(j, j^{'})$ of the considered node such that $j, j^{'} \in \{1, 2, \dots, M\}$ and $j \neq j^{'}$. Thus when splitting any node, its label histogram is divided in such a way that its children have non-overlapping label histograms, i.e. $\forall_{i=1,2,\dots,K}\forall_{j,j'\in\{1,2,\dots,M\},j\neq j'}\rho_i^{(j)}\rho_i^{(j^{'})} = 0$, where $\rho_i^{(j)}$ and $\rho_i^{(j^{'})}$ denote the i^{th} entry in the normalized label histograms of children nodes j and $j^{'}$ respectively. After $\log_M(K/r)$ splits we obtain leaves with non-overlapping histograms, i.e. for any two leaves l_1 and l_2 such that $l_1, l_2 \in \mathcal{L}$ and $l_1 \neq l_2$, $\forall_{i=1,2,\dots,K}\rho_i^{(l_1)} \cdot \rho_i^{(l_2)} = 0$. In each leaf the label histogram contains r non-zero entries. Based on the above it follows that $G(\mathcal{T}) = 0$. Consequently, using Equation 67 we obtain that the multi-label error $\epsilon_r(\mathcal{T})$ is equal to zero as well. This directly implies that $\epsilon_{\hat{r}}(\mathcal{T}) = 0$ for any $\hat{r} = 1, 2, \dots, r$. $\square$

Proof of Lemma 8 (Proof of Lemma 5 follows directly as Lemma 5 is a special case of Lemma 8). In our algorithm we store label histograms for each node, and at testing we assign to an example top r labels obtained from averaging the histograms of the leaves to which this example has descended to. At training, we recursively find the node with the highest priority and partition it to two children. Here we are examining the change of error with one node split. We consider examples reaching that node and without loss of generality we assume they have reached only this node. For each such example x we assign the top r labels from the histogram of the analyzed node, i.e. $y_r(x) = \{k_1, k_2, ..., k_r\}$, where $k_1 = \text{argmax}_{k\in\{1,2,\dots,K\}}\rho_k$, $k_2 = \text{argmax}_{k\in\{1,2,\dots,K\}\backslash j_1}(\rho_k)$,..., $k_r = \text{argmax}_{k\in\{1,2,\dots,K\}\backslash\{j_1,\dots,j_{r-1}\}}(\rho_k)$ and ρ_i is the probability that the data point x has label i given that x has reached node n, i.e. $\rho_i = P(i \in t(x)|x \text{ reached } n)$. After t splits the Precision can be expanded as follows:

$$\begin{aligned}
(P@r)^t &= \frac{1}{r}\sum_{i=1}^{K} P(i \in t(x), i \in y_r(x)) && (73)\\
&= \frac{1}{r}\Big(\max_{k\in\{1,2,\dots,K\}}\rho_k + \max_{k\in\{1,2,\dots,K\}\backslash j_1}\rho_k + \dots + \max_{k\in\{1,2,\dots,K\}\backslash\{j_1,j_2,\dots,j_{r-1}\}}\rho_k\Big) && (74)\\
&= \max_{k\in\{1,2,\dots,K\}}\pi_k + \max_{k\in\{1,2,\dots,K\}\backslash j_1}\pi_k + \dots + \max_{k\in\{1,2,\dots,K\}\backslash\{j_1,j_2,\dots,j_{r-1}\}}\pi_k && (75)\\
&= \pi_{k_1} + \dots + \pi_{k_r}, && (76)
\end{aligned}$$

where the last line comes from the fact that π_i is a normalized fraction of examples containing label i in their labels. After the node split, the Precision is defined as the combination of the Precision of its children. For simplicity we consider equal contribution of each of the edges to $P_{\text{multi}} = \left|\left(\sum_{j=1}^{M} P_j\right) - 1\right|$. Therefore we can write the Precisions of the children as:

$$\begin{aligned}
(P@r)^{t+1} &= (P_1 - \frac{1}{M}P_{\text{multi}})(P@r)^1 + \cdots + (P_M - \frac{1}{M}P_{\text{multi}})(P@r)^M && (77)\\
&= (P_1 - \frac{1}{M}P_{\text{multi}})\Big(\max_{i\in\{1,2,\ldots,K\}} \pi_i\big(\frac{P_1^i - \frac{1}{M}P^i_{\text{multi}}}{P_1 - \frac{1}{M}P_{\text{multi}}}\big) + \cdots\Big) + \cdots && (78)\\
&+ (P_M - \frac{1}{M}P_{\text{multi}})\Big(\max_{j\in\{1,2,\ldots,K\}} \pi_j\big(\frac{P_M^j - \frac{1}{M}P^j_{\text{multi}}}{P_M - \frac{1}{M}P_{\text{multi}}}\big) + \cdots\Big)\\
&= \max_{i\in\{1,2,\ldots,K\}} \pi_i(P_1^i - \frac{1}{M}P^i_{\text{multi}}) + \cdots && (79)\\
&+ \max_{j\in\{1,2,\ldots,K\}} \pi_j(P_M^j - \frac{1}{M}P^j_{\text{multi}}) + \cdots\\
&= \frac{1}{M}\big(\max_{i\in\{1,2,\ldots,K\}} \pi_i((M-1)P_1^i - P_2^i \cdots - P_M^i + 1) + \cdots && (80)\\
&+ \max_{j\in\{1,2,\ldots,K\}} \pi_j((M-1)P_M^i - P_1^i \cdots - P_{M-1}^i + 1) + \cdots\big)\\
&= \frac{1}{M}\big(\max_{i\in\{1,2,\ldots,K\}} \pi_i((P_1^i - P_2^i) + (P_1^i - P_3^i) + \cdots (P_1^i - P_M^i) + 1) + \cdots && (81)\\
&+ \max_{j\in\{1,2,\ldots,K\}} \pi_j((P_M^i - P_1^i) + (P_M^i - P_2^i) + \cdots (P_M^i - P_{M-1}^i) + 1) + \cdots\big).
\end{aligned}$$

Note that the subtraction of $(1/M)P^i_{\text{multi}}$ and $(1/M)P_{\text{multi}}$ in the coefficients is done to compensate the Precision calculation for examples being sent to multiple directions. Let the top r labels assigned to the first child be denoted as $y_r^1(x) = \{i_1, i_2, ..., i_r\}$, where
$i_1 = \text{argmax}_{i\in\{1,2,\ldots,K\}}\pi_i((P_1^i - P_2^i) + (P_1^i - P_3^i) + \cdots (P_1^i - P_M^i))$,
$i_2 = \text{argmax}_{k\in\{1,2,\ldots,K\}\setminus i_1}\pi_i((P_1^i - P_2^i) + (P_1^i - P_3^i) + \cdots (P_1^i - P_M^i))$,
...,
$i_r = \text{argmax}_{k\in\{1,2,\ldots,K\}\setminus\{i_1,\ldots,i_{r-1}\}}\pi_i((P_1^i - P_2^i) + (P_1^i - P_3^i) + \cdots (P_1^i - P_M^i))$.
Analogy holds for all other children. Thus for example the M^{th} children's labels are: $y_r^M(x) = \{j_1, j_2, ..., j_r\}$. Therefore the difference between the Precision of the parent node and its children can be written as:

$$\begin{aligned}
(P@r)^{t+1} - (P@r)^t = \frac{1}{M}\Big(&\pi_{i_1}((P_1^{i_1} - P_2^{i_1}) + \cdots (P_1^{i_1} - P_M^{i_1}) + 1) + \cdots && (82)\\
&+ \pi_{i_r}((P_1^{i_r} - P_2^{i_r}) + \cdots (P_1^{i_r} - P_M^{i_r}) + 1)\Big)\\
&+ \cdots\\
&+ \frac{1}{M}\Big(\pi_{j_1}((P_M^{j_1} - P_1^{j_1}) + \cdots (P_M^{j_1} - P_{M-1}^{j_1}) + 1) + \cdots\\
&+ \pi_{j_r}((P_M^{j_r} - P_1^{j_r}) + \cdots (P_M^{j_r} - P_{M-1}^{j_r}) + 1)\Big)\\
&- \big(\pi_{k_1} + \cdots + \pi_{k_r}\big).
\end{aligned}$$

For the ease of notation we show the case for the binary below:

$$\begin{aligned}
(P@r)^{t+1} - (P@r)^t = &\frac{1}{2}\big(\pi_{i_1}(P_R^{i_1} - P_L^{i_1} + 1) + \cdots + \pi_{i_r}(P_R^{i_r} - P_L^{i_r} + 1)\big) && (83)\\
&+ \frac{1}{2}\big(\pi_{j_1}(P_L^{j_1} - P_R^{j_1} + 1) + \cdots + \pi_{j_r}(P_L^{j_r} - P_R^{j_r} + 1)\big)\\
&- \big(\pi_{k_1} + \cdots + \pi_{k_r}\big).
\end{aligned}$$

Considering the Assumption 3.1,we have at least one label such that $P_R^k - P_L^k = \gamma_1 > 0, \gamma_1 \in (0,1]$. Without loss of generality let $P_R^{k_1} - P_L^{k_1} = \gamma_1 > 0$ for the top label in the parent node. Thus: $\pi_{i_1}(P_R^{i_1} - P_L^{i_1} + 1) \geq \pi_{k_1}(1+\gamma_1)$ and $\pi_{j_1}(P_L^{j_1} - P_R^{j_1} + 1) \geq \pi_{k_1}(1-\gamma_1)$. Therefore we have $(P@r)^{t+1} - (P@r)^t \geq 0$. Due to the weak hypothesis assumption the histograms in the children nodes are different than in the parent on at least one position corresponding to one label. If that label is in the top r labels that we assign to the children node, the error will be reduced. If not, the error is going to be the same, but that cannot happen forever, i.e. for some split the label(s) for which the weak hypothesis assumption holds will eventually be in the top r labels that are assigned to the children node. To put this intuition into more formal language, if any of the top r labels in any of the children are different from the top r parent labels, i.e. $y_r^1 \neq y_r$, $y_r^2 \neq y_r$,..., or $y_r^M \neq y_r$ we will have $(P@r)^{t+1} - (P@r)^t > 0$. Because of the weak hypothesis assumption, the latter condition is inevitable and will eventually hold after some node split. This shows that the error is monotonically decreasing. □

9 ADDITIONAL ALGORITHMS

Algorithm 3 OptimizeObjective (v)

```
J_opt ← +∞
for s = 1 ... 2^M − 1 do
    for m = 1 ... M do
        ŷ[m] = s ∧ 2^(m−1) > 0
        P_m ← ((v.C_v − y_i.size()) v.P_m + y_i.size() * ŷ[m]) / v.C_v
        for k ∈ y_i do
            P_m^k ← ((v.l_v[k] − 1) v.P_m^k + ŷ[m]) / v.l_v[k]
        end for
    end for
    % objective computation
    B ← Σ_{j=1}^{M} Σ_{l=j+1}^{M} |P_j − P_l|
    CI ← Σ_{i=1}^{y_i.size()} Σ_{j=1}^{M} Σ_{l=j+1}^{M} (v.l_v(i) / v.C_v) |P_j^i − P_l^i|
    MWP ← |(Σ_{j=1}^{M} P_j) − 1|
    J ← B − λ_1 CI + λ_2 MWP
    if J < J_opt then
        J_opt ← J
        ŷ_opt ← ŷ
    end if
end for
return ŷ_opt
```

Algorithm 4 TrainRegressors (v)

```
% y_i.size() denotes the size of vector y_i
v.C_v ← 0;   v.l_v ← ∅;   v.isLeaf ← false
for m = 1 ... M do
    v.w_m ← random weights;   v.P_m ← 0
    for i=1...K do  v.P_m^i ← 0  end for
end for
for e = 1 ... E do
    for i ∈ v.I do
        for k ∈ y_i do
            v.C_v++;   v.l_v[k]++
        end for
        ŷ ← OptimizeObjective (v)
        for m = 1 ... M do
            Train v.w_m with example (x_i, ŷ[m])
            pred ← clamp_[0,1](v.w_m^T x_i)
            v.P_m ← ((v.C_v − y_i.size())) * v.P_m + y_i.size() * pred) / v.C_v
            for k ∈ y_i do
                v.P_m^k ← ((v.l_v[k] − 1) * v.P_m^k + pred) / v.l_v[k]
            end for
        end for
    end for
end for
```

Algorithm 5 CreateChildren (v)

```
for m = 1 ... M do
    v.ch[m].I ← ∅
    v.ch[m].Lhist ← ∅
    v.ch[m].isLeaf ← true
end for
for i ∈ v.I do
    sent ← false
    for m ∈ 1 ... M do
        if v.w_m^⊤ x_i > 0.5 then
            % example (x_i, y_i) goes to child m
            UpdateHist (v.ch[m].Lhist, y_i)
            v.ch[m].I.push(i)
            sent ← true
        end if
    end for
    if not sent then
        m ← arg max_{m̂∈{1,2,...,M}} v.w_m̂^⊤ x_i
        UpdateHist (v.ch[m].Lhist, y_i)
        v.ch[m].I.push(i)
    end if
end for
return v.ch
```

10 EXPERIMENTAL SETUP

LdSM was implemented in C++. The regressors in the tree nodes were trained with either SGD [Bottou (1998)] (Mediamill) or NAG [Ross et al. (2013)] (remaining data sets) with step size chosen from $[0.001, 1]$. The trees were trained with up to 20 passes through the data and we explored trees with up to $64K$ nodes for Mediamill and Bibtex, up to $32K$ for Delicious, and up to $2K$ for the rest of the data sets. λ_1 and λ_2 were chosen from the set $\{0.5, 1, 1.5, 2, 4\}$ and M was set to either 2 or 4. FastXML, PFastreXML, CRAFTML and LdSM algorithms use tree ensembles of size ~ 50. PLT and LPSR use a single tree, and GBDT-S uses up to 100 trees.

Table 5: Data set statistics.

Data Sets	#Features	#Labels	#Training samples	#Testing samples	Avg. Labels per Point	Avg. Points per Label
Mediamill	120	101	30993	12914	4.38	1902.15
Bibtex	1836	159	4880	2515	2.40	111.71
Delicious	500	983	12920	3185	19.03	311.61
Eurlex	5000	3993	15539	3809	5.31	25.73
AmazonCat-13k	203882	13330	1186239	306782	5.04	448.57
Wiki10-31k	101938	30938	14146	6616	18.64	8.52
Delicious-200k	782585	205443	196606	100095	75.54	72.29
Amazon-670k	135909	670091	490449	153025	5.45	3.99

Table 6: Experimental setup that was used to obtain results for various data sets with LdSM method: the depth of the deepest tree in the ensemble and tree arity.

Data sets	Depth	Arity
Mediamill	9	4
Bibtex	9	4
Delicious	10	4
AmazonCat-13k	18	2
Wiki10-31k	10	4
Delicious-200k	46	2
Amazon-670k	25	2

11 ADDITIONAL EXPERIMENTAL RESULTS

Table 7: Prediction time [ms] per example for tree-based approaches: GBDT-S, CRAFTML, FastXML, PFastreXML, LdSM (LPSR and PLT are NA) and other (not purely tree-based) methods: Parabel, DisMEC Babbar and Schölkopf (2017), PD-Sparse Yen et al. (2016), PPD-Sparse Yen et al. (2017), OVA-Primal++ H. Fang and Friedlander (2019) and SLEEC Bhatia et al. (2015) on various data sets. The best result among tree-based methods is in bold, and among all methods is underlined.

	Tree-based				
	GBDT-S	CRAFTML	FastXML	PFastreXML	LdSM
Mediamill	0.05	NA	0.27	0.37	**0.05**
Bibtex	NA	NA	0.64	0.73	**0.013**
Delicious	0.04	NA	NA	NA	<u>**0.014**</u>
AmazonCat-13k	NA	5.12	1.21	1.34	<u>**0.04**</u>
Wiki10-31k	0.20	NA	1.38	NA	<u>**0.15**</u>
Delicious-200k	**0.14**	8.6	1.28	7.40	1.21
Amazon-670k	NA	5.02	1.48	1.98	<u>**0.12**</u>

	Other					
	Parabel	DiSMEC	PD-Sparse	PPD-Sparse	OVA-Primal++	SLEEC
Mediamill	NA	0.142	<u>0.004</u>	0.078	NA	4.95
Bibtex	NA	0.28	<u>0.007</u>	0.094	NA	0.70
Delicious	NA	NA	NA	NA	NA	NA
AmazonCat-13k	NA	0.20	0.87	1.82	NA	13.36
Wiki10-31k	NA	116.66	NA	NA	NA	NA
Delicious-200k	NA	311.4	<u>0.43</u>	275	NA	2.69
Amazon-670k	1.13	148	NA	20	NA	6.94

Table 8: Training time [s] for tree-based approaches: GBDT-S, CRAFTML, FastXML, PFastreXML, LdSM (LPSR and PLT are NA) and other (not purely tree-based) methods: Parabel, DisMEC, PD-Sparse, PPD-Sparse, SLEEC, on various data sets. The best result among tree-based methods is in bold, and among all methods is underlined.

	Tree-based				
	GBDT-S	CRAFTML	FastXML	PFastreXML	LdSM
Mediamill	NA	NA	276.4	293.2	**52.7**
Bibtex	NA	NA	21.68	21.47	**9.48**
Delicious	NA	NA	NA	NA	<u>**21.74**</u>
AmazonCat-13k	NA	2876	11535	13985	**607**
Wiki10-31k	1044	NA	1275.9	NA	<u>**179**</u>
Delicious-200k	NA	<u>**1174**</u>	8832.46	8807.51	5125
Amazon-670k	NA	1487	5624	6559	**957**

	Other					
	Parabel	DiSMEC	PD-Sparse	PPD-Sparse	OVA-Primal++	SLEEC
Mediamill	NA	<u>12.15</u>	34.1	23.8	NA	9504
Bibtex	NA	<u>0.203</u>	7.71	0.232	NA	296.86
Delicious	NA	NA	NA	NA	NA	NA
AmazonCat-13k	NA	11828	2789	<u>122.8</u>	7330	119840
Wiki10-31k	NA	NA	NA	NA	1364	NA
Delicious-200k	NA	38814	5137.4	2869	NA	4838.7
Amazon-670k	1512	174135	NA	<u>921.9</u>	NA	20904

Remark 3 (Training time)**.** *The training time of LdSM can be reduced order of magnitudes by using lower number of epochs at the expense of* $\sim 1\%$ *loss in the accuracy. However, we report the training times that correspond to the best accuracy results obtained with LdSM.*

Table 9: Propensity Score Precisions: $PSP@1$, $PSP@3$, and $PSP@5$ (%) and Propensity Score nDCG scores: $PSN@1$, $PSN@3$, and $PSN@5$ (%) obtained by different tree-based methods on common multi-label data sets.

Mediamill $D = 120, K = 101$						
Algorithm	PSP@1	PSP@3	PSP@5	PSN@1	PSN@3	PSN@5
LPSR	66.06	63.83	61.11	66.06	64.83	62.94
FastXML	66.67	65.43	64.30	66.67	66.08	65.24
PFastreXML	66.88	65.90	64.90	66.88	66.47	65.71
LdSM	**70.27**	**69.66**	**68.86**	**70.27**	**69.99**	**70.30**

Bibtex $D = 1.8k, K = 159$						
Algorithm	PSP@1	PSP@3	PSP@5	PSN@1	PSN@3	PSN@5
LPSR	49.20	50.14	55.01	49.20	49.78	52.41
FastXML	48.54	52.30	58.28	48.54	51.11	54.38
PFastreXML	**52.28**	54.36	**60.55**	**52.28**	53.62	56.99
LdSM	52.01	**54.38**	60.34	52.01	**53.67**	**57.08**

Delicious $D = 500, K = 983$						
Algorithm	PSP@1	PSP@3	PSP@5	PSN@1	PSN@3	PSN@5
LPSR	31.34	32.57	32.77	31.34	32.29	32.50
FastXML	32.35	34.51	35.43	32.35	34.00	34.73
PFastreXML	34.57	34.80	35.86	34.57	34.71	35.42
LdSM	**37.27**	**38.32**	**38.46**	**37.27**	**38.09**	**38.28**

AmazonCat-13k $D = 204k, K = 13k$						
Algorithm	PSP@1	PSP@3	PSP@5	PSN@1	PSN@3	PSN@5
LPSR	-	-	-	-	-	-
FastXML	48.31	60.26	69.30	48.31	56.90	62.75
PFastreXML	**69.52**	**73.22**	**75.48**	**69.52**	**72.21**	**73.67**
LdSM	51.06	58.67	60.47	51.06	57.78	60.52

Wiki10-31k $D = 102k, K = 31k$						
Algorithm	PSP@1	PSP@3	PSP@5	PSN@1	PSN@3	PSN@5
LPSR	12.79	12.26	12.13	12.79	12.38	12.27
FastXML	9.80	10.17	10.54	9.80	10.08	10.33
PFastreXML	**19.02**	**18.34**	**18.43**	**19.02**	**18.49**	**18.52**
LdSM	11.87	12.35	12.89	11.87	12.42	12.58

Delicious-200k $D = 783k, K = 205k$						
Algorithm	PSP@1	PSP@3	PSP@5	PSN@1	PSN@3	PSN@5
LPSR	3.24	3.42	3.64	3.24	3.37	3.52
FastXML	6.48	7.52	8.31	6.51	7.26	7.79
PFastreXML	3.15	3.87	4.43	3.15	3.68	4.06
LdSM	**7.16**	**8.26**	**9.11**	**7.16**	**7.92**	**8.45**

Amazon-670k $D = 135k, K = 670k$						
Algorithm	PSP@1	PSP@3	PSP@5	PSN@1	PSN@3	PSN@5
LPSR	16.68	18.07	19.43	16.68	17.70	18.63
FastXML	19.37	23.26	26.85	19.37	22.25	24.69
PFastreXML	**29.30**	30.80	32.43	**29.30**	**30.40**	**31.49**
LdSM	28.14	**30.82**	**33.16**	28.14	29.80	30.71

Table 10: Precisions: $P@1$, $P@3$, and $P@5$ (%) and nDCG scores: $N@1$, $N@3$, and $N@5$ (%) obtained for tree-based approaches: GBDT-S, CRAFTML, FastXML, PFastreXML, LPSR, PLT, and LdSM and other (not purely tree-based) methods: Parabel, DisMEC, PD-Sparse, PPD-Sparse, OVA-Primal++, LEML, and SLEEC, on various data sets. The best result among tree-based methods is in bold, and among all methods is underlined.

Mediamill

	Algorithm	P@1	P@3	P@5	N@1	N@3	N@5
Other	Parabel	83.91	67.12	52.99	83.91	75.22	72.21
	DiSMEC	-	-	-	-	-	-
	PD-Sparse	81.86	62.52	45.11	81.86	70.21	63.71
	PPD-Sparse	-	-	-	-	-	-
	OVA-Primal	-	-	-	-	-	-
	LEML	84.01	67.20	52.80	84.01	75.23	71.96
	SLEEC	87.82	73.45	59.17	87.82	81.50	79.22
Tree	LPSR	83.57	65.78	49.97	83.57	74.06	69.34
	PLT	-	-	-	-	-	-
	GBDT-S	84.23	67.85	-	-	-	-
	CRAFTML	85.86	69.01	54.65	-	-	-
	FastXML	84.22	67.33	53.04	84.22	75.41	72.37
	PFastreXML	83.98	67.37	53.02	83.98	75.31	72.21
	LdSM	**90.64**	**73.60**	**58.62**	**90.64**	**82.14**	**79.23**

Bibtex

	Algorithm	P@1	P@3	P@5	N@1	N@3	N@5
Other	Parabel	64.53	38.56	27.94	64.53	59.35	61.06
	DiSMEC	-	-	-	-	-	-
	PD-Sparse	61.29	35.82	25.74	61.29	55.83	57.35
	PPD-Sparse	-	-	-	-	-	-
	OVA-Primal	-	-	-	-	-	-
	LEML	62.54	38.41	28.21	62.54	58.22	60.53
	SLEEC	65.08	39.64	28.87	65.08	60.47	62.64
Tree	LPSR	62.11	36.65	26.53	62.11	56.50	58.23
	PLT	-	-	-	-	-	-
	GBDT-S	-	-	-	-	-	-
	CRAFTML	**65.15**	**39.83**	28.99	-	-	-
	FastXML	63.42	39.23	28.86	63.42	59.51	61.70
	PFastreXML	63.46	39.22	29.14	63.46	59.61	62.12
	LdSM	64.69	39.70	**29.25**	**64.69**	**60.37**	**62.73**

Delicious

	Algorithm	P@1	P@3	P@5	N@1	N@3	N@5
Other	Parabel	67.44	61.83	56.75	67.44	63.15	59.41
	DiSMEC	-	-	-	-	-	-
	PD-Sparse	51.82	44.18	38.95	51.82	46.00	42.02
	PPD-Sparse	-	-	-	-	-	-
	OVA-Primal	-	-	-	-	-	-
	LEML	65.67	60.55	56.08	65.67	61.77	58.47
	SLEEC	67.59	61.38	56.56	67.59	62.87	59.28
Tree	LPSR	65.01	58.96	53.49	65.01	60.45	56.38
	PLT	-	-	-	-	-	-
	GBDT-S	69.29	63.62	-	-	-	-
	CRAFTML	70.26	63.98	59.00	-	-	-
	FastXML	69.61	64.12	59.27	69.61	65.47	61.90
	PFastreXML	67.13	62.33	58.62	67.13	63.48	60.74
	LdSM	**71.91**	**65.34**	**60.24**	**71.91**	**66.90**	**63.09**

AmazonCat-13k

	Algorithm	P@1	P@3	P@5	N@1	N@3	N@5
Other	Parabel	93.03	79.16	64.52	93.03	87.72	86.00
	DiSMEC	93.40	79.10	64.10	93.40	87.70	85.80
	PD-Sparse	90.60	75.14	60.69	90.60	84.00	82.05
	PPD-Sparse	-	-	-	-	-	-
	OVA-Primal	93.75	78.89	63.66	-	-	-
	LEML	-	-	-	-	-	-
	SLEEC	90.53	76.33	61.52	90.53	84.96	82.77
Tree	LPSR	-	-	-	-	-	-
	PLT	91.47	75.84	61.02	-	-	-
	GBDT-S	-	-	-	-	-	-
	CRAFTML	92.78	**78.48**	63.58	-	-	-
	FastXML	93.11	78.2	63.41	93.11	**87.07**	**85.16**
	PFastreXML	91.75	77.97	**63.68**	91.75	86.48	84.96
	LdSM	**93.87**	75.41	57.86	**93.87**	85.06	80.63

Wiki10-31k

	Algorithm	P@1	P@3	P@5	N@1	N@3	N@5
Other	Parabel	84.31	72.57	63.39	83.03	71.01	68.30
	DiSMEC	85.20	74.60	65.90	84.10	77.10	70.40
	PD-Sparse	-	-	-	-	-	-
	PPD-Sparse	-	-	-	-	-	-
	OVA-Primal	84.17	74.73	65.92	-	-	-
	LEML	73.47	62.43	54.35	73.47	64.92	58.69
	SLEEC	85.88	72.98	62.70	85.88	76.02	68.13
Tree	LPSR	72.72	58.51	49.50	72.72	61.71	54.63
	PLT	84.34	72.34	62.72	-	-	-
	GBDT-S	84.34	70.82	-	-	-	-
	CRAFTML	**85.19**	**73.17**	**63.27**	-	-	-
	FastXML	83.03	67.47	57.76	83.03	**75.35**	63.36
	PFastreXML	83.57	68.61	59.10	83.57	72.00	64.54
	LdSM	83.74	71.74	61.51	**83.74**	74.60	**66.77**

Delicious-200k

	Algorithm	P@1	P@3	P@5	N@1	N@3	N@5
Other	Parabel	46.97	40.08	36.63	46.97	41.72	39.07
	DiSMEC	45.50	38.70	35.50	45.50	40.90	37.80
	PD-Sparse	34.37	29.48	27.04	34.37	30.60	28.65
	PPD-Sparse	-	-	-	-	-	-
	OVA-Primal	-	-	-	-	-	-
	LEML	40.73	37.71	35.84	40.73	38.44	37.01
	SLEEC	47.85	42.21	39.43	47.85	43.52	41.37
Tree	LPSR	18.59	15.43	14.07	18.59	16.17	15.13
	PLT	45.37	38.94	35.88	-	-	-
	GBDT-S	42.11	39.06	-	-	-	-
	CRAFTML	**47.87**	**41.28**	38.01	-	-	-
	FastXML	43.07	38.66	36.19	43.07	39.70	37.83
	PFastreXML	41.72	37.83	35.58	41.72	38.76	37.08
	LdSM	45.26	40.53	**38.23**	**45.26**	**41.66**	**39.79**

Amazon-670k

	Algorithm	P@1	P@3	P@5	N@1	N@3	N@5
Other	Parabel	44.89	39.80	36.00	44.89	42.14	40.36
	DiSMEC	44.70	39.70	36.10	44.70	42.10	40.50
	PD-Sparse	-	-	-	-	-	-
	PPD-Sparse	45.32	40.37	36.92	-	-	-
	OVA-Primal	-	-	-	-	-	-
	LEML	8.13	6.83	6.03	8.13	7.30	6.85
	SLEEC	35.05	31.25	28.56	34.77	32.74	31.53
Tree	LPSR	28.65	24.88	22.37	28.65	26.40	25.03
	PLT	36.65	32.12	28.85	-	-	-
	GBDT-S	-	-	-	-	-	-
	CRAFTML	37.35	33.31	30.62	-	-	-
	FastXML	36.99	33.28	30.53	36.99	35.11	33.86
	PFastreXML	39.46	35.81	33.05	39.46	37.78	36.69
	LdSM	**42.63**	**38.09**	**34.70**	**42.63**	**40.37**	**38.89**

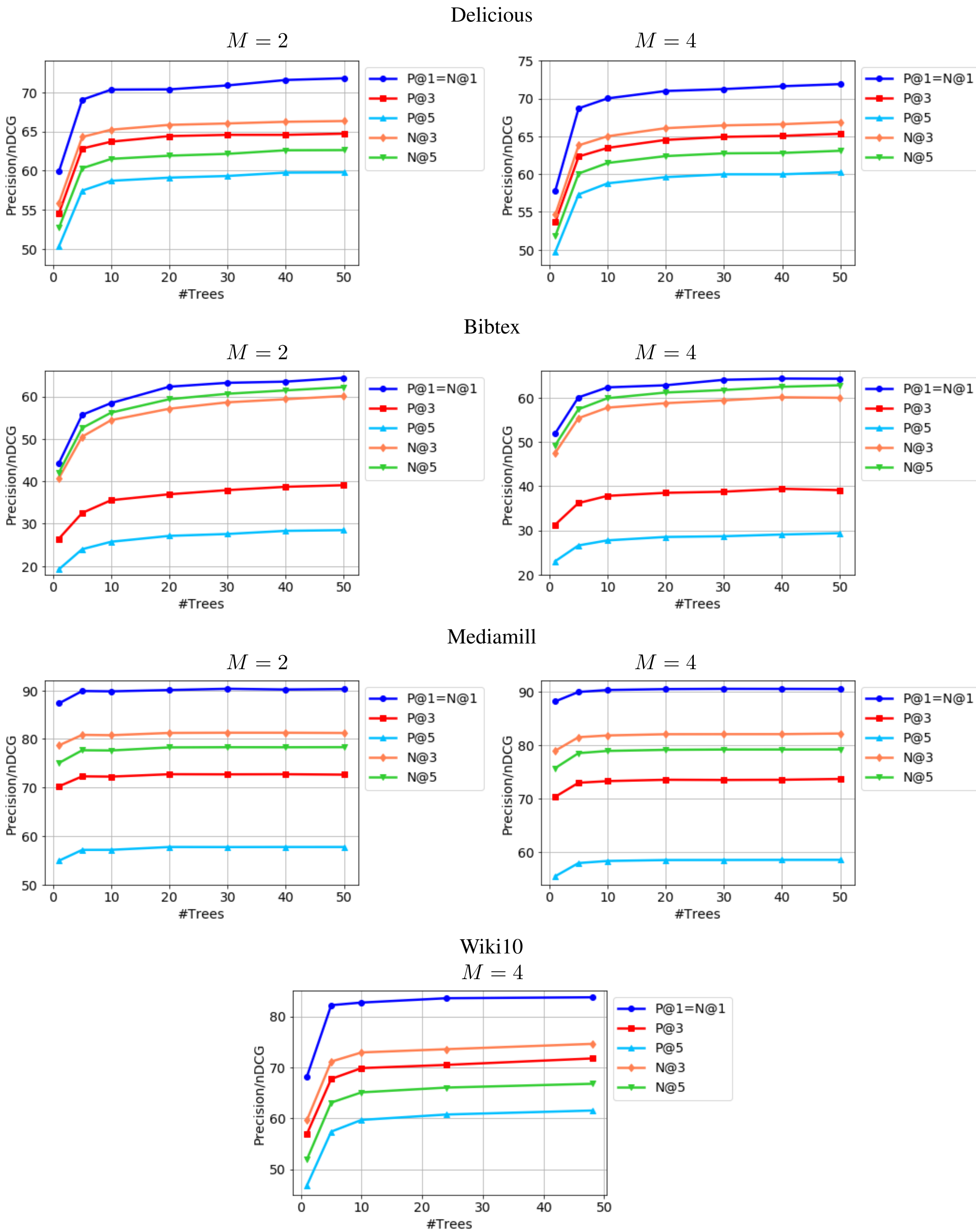

Figure 4: The behavior of Precision/nDCG score as a function of the number of trees in the ensemble. Plots were obtained for Delicious, Bibtex, Mediamill, and Wiki10 data sets.

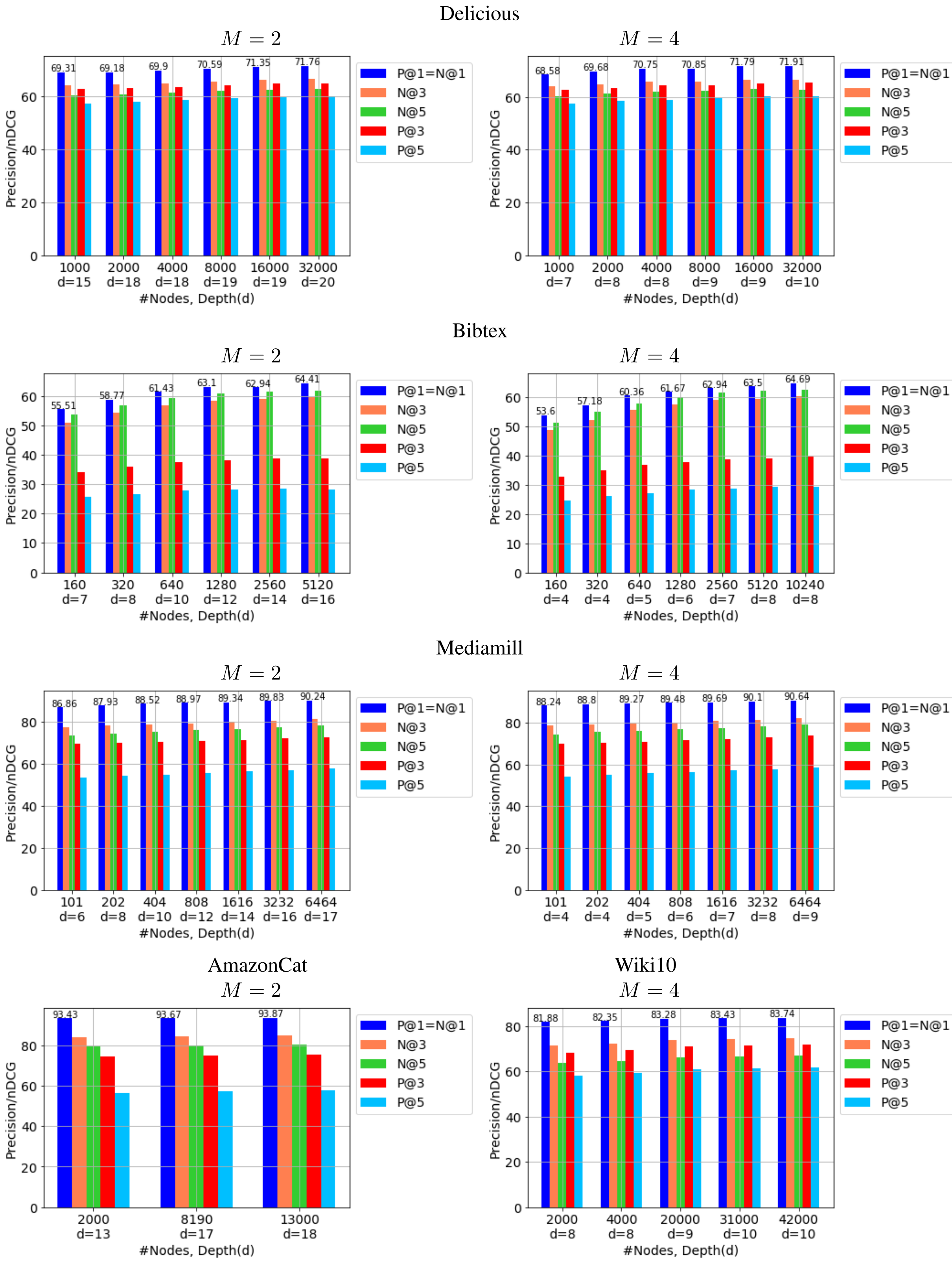

Figure 5: The behavior of Precision/nDCG score as a function of the number of nodes T_{max} (including leaves) and tree depth of the deepest tree in the ensemble. Plots were obtained for Delicious, Bibtex, Mediamill, AmazonCat, and Wiki10 data sets.

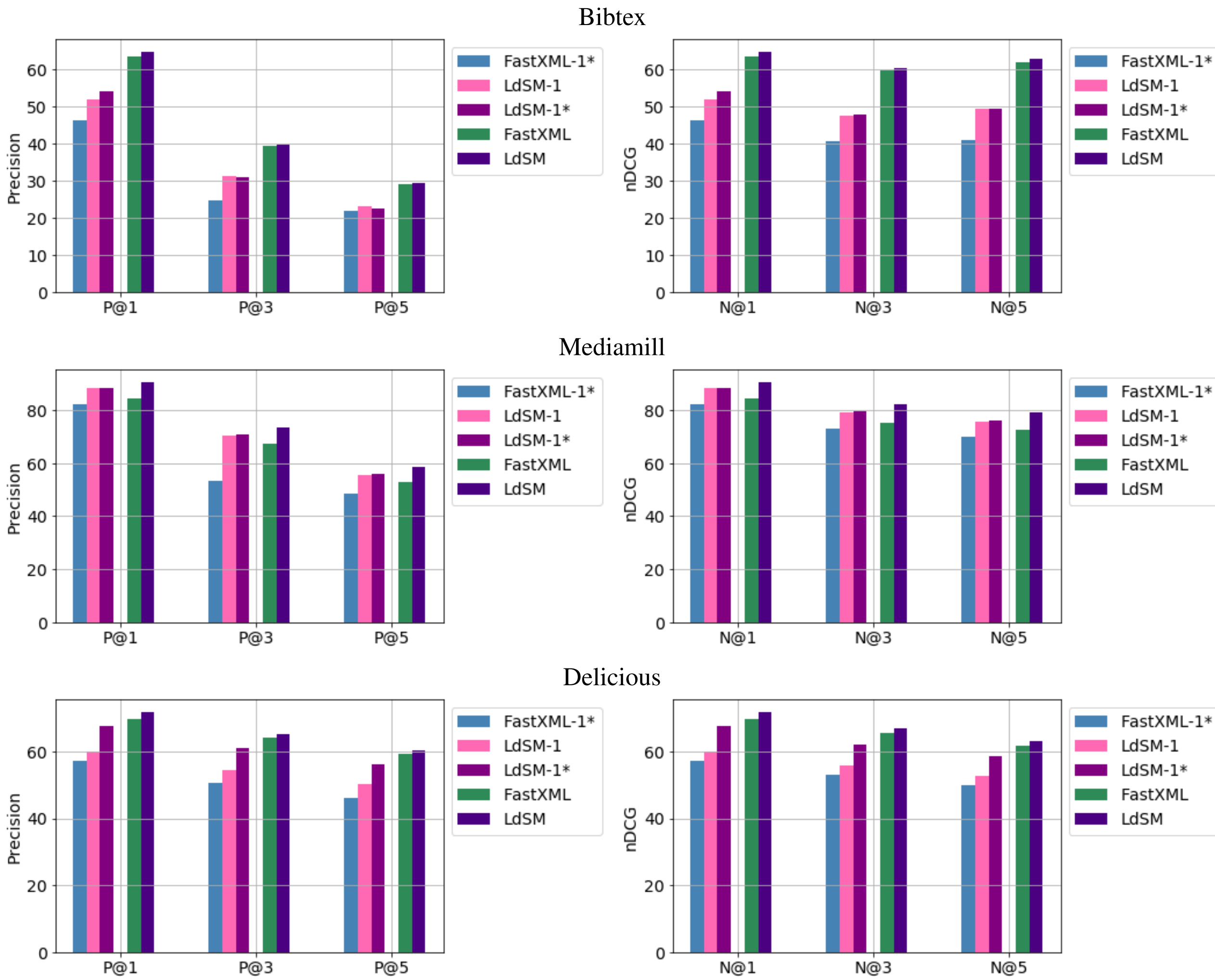

Figure 6: The comparison of Precision (**left column**) and nDCG (**right column**) score for LdSM and FastXML working in the ensemble (**right bars**) as well as for single-tree (**left bars**) (LdSM-1: exemplary tree chosen from LdSM ensemble, LdSM-1*, FastXML-1*: optimal single trees). Plots were obtained for Bibtex, Mediamill and Delicious data sets.