A. Proof of theorem 2

Proof. For the fixed tree at timestep t (there have been $t-1$ previous splits) with a fixed partition function in the nodes, the weighted entropy of class labels is $W_t = \sum_{\{n \in \text{Leaves}\}} p_n H_n$.

When we split the tth node, the weak learning assumption implies entropy decreases by γ according to:

$$H_n \geq \left(\frac{p_l}{p_n} H_l + \frac{p_r}{p_n} H_r \right) + \gamma$$

where γ is the advantage of the weak learner. Hence,

$$W_t - W_{t+1} = p_n H_n - p_l H_l - p_r H_r \geq p_n \gamma$$

We can bound p_n according to

$$\max_n p_n \geq \frac{1}{t}$$

which implies

$$W_t - W_{t+1} \geq \frac{\gamma}{t}$$

This can be solved recursively to get:

$$\begin{aligned} W_{t+1} &\leq W_1 - \gamma \sum_{i=1}^{t} \frac{1}{i} \\ &\leq W_1 - \gamma \ln(t+1) \\ &= H_1 - \gamma \ln(t+1) \end{aligned}$$

where the second inequality follows from lower bounding a harmonic series, and H_1 is the marginal Shannon entropy of the class labels.

To finish the proof, we bound the multiclass loss in terms of the average entropy. For any leaf node n we can assign the most likely label, $y = \arg\max_i \pi_{ni}$ so the error rate is $\epsilon_n = 1 - \pi_{ny}$.

$$\begin{aligned} W_{t+1} &= \sum_{\{n \in \text{Leaves}\}} p_n \sum_i \pi_{ni} \ln \frac{1}{\pi_{ni}} \\ &\geq \sum_{\{n \in \text{Leaves}\}} p_n \sum_i \pi_{ni} \ln \frac{1}{\pi_{ny}} \\ &= \sum_{\{n \in \text{Leaves}\}} p_n \ln \frac{1}{1 - \epsilon_n} \\ &\geq \sum_{\{n \in \text{Leaves}\}} p_n \epsilon_n \\ &= \epsilon \end{aligned}$$

Putting these inequalities together we have:

$$\epsilon \leq H_1 - \gamma \ln(t+1)$$

□

Algorithm	Parameter	Default Value
Binary	Learning Rate	1
	Loss	logistic
Recall Tree	Max Depth	$\log_2$(#classes)
	Num Candidates	$4\log_2$(#classes)
	Depth Penalty (λ)	1

Table 3. Algorithm hyperparameters for various algorithms. "Binary" refers to hyperparameters inherited via reduction to binary classification.

B. Datasets

ALOI (Geusebroek et al., 2005) is a color image collection of one-thousand small objects recorded for scientific purposes (Geusebroek et al., 2005). We use the same train-test split and representation as Choromanska et. al. (Choromanska & Langford, 2015).

Imagenet consists of features extracted from intermediate layers of a convolutional neural network trained on the ILVSRC2012 challenge dataset. This dataset was originally developed to study transfer learning in visual tasks (Oquab et al., 2014); more details are at `http://www.di.ens.fr/willow/research/cnn/`. We utilize a predictor linear in this representation.

LTCB is the Large Text Compression Benchmark, consisting of the first billion bytes of a particular Wikipedia dump (Mahoney, 2009). Originally developed to study text compression, it is now commonly used as a language modeling benchmark where the task is to predict the next word in the sequence. We limit the vocabulary to 80000 words plus a single out-of-vocabulary indicator; utilize a model linear in the 6 previous unigrams, the previous bigram, and the previous trigram; and utilize a 90-10 train-test split on entire Wikipedia articles.

ODP(Bennett & Nguyen, 2009) is a multiclass dataset derived from the Open Directory Project. We utilize the same train-test split and labels from (Choromanska & Langford, 2015). Specifically there is a fixed train-test split of 2:1, the representation of a document is a bag of words, and the class label is the most specific category associated with each document.

C. Experimental Methodology

Default Performance Methodology Hyperparameter selection can be computationally burdensome for large data sets, which is relevant to any claims of decreased training times. Therefore we report results using the default values indicated in Table 3. For the larger data sets (Imagenet, ODP), we do a single pass over the training data;

Table 2. Empirical comparison summary. When OAA training is accelerated using parallelism and gradient subsampling, wall clock times are parenthesized. Training times are for defaults, i.e., without hyperparameter optimization. Asterisked LOMTree results are from (Choromanska & Langford, 2015).

Dataset	Method	Test Error		Training Time	Inference Time per example
		Default	Tuned		
ALOI	OAA	12.2%	12.1%	571s	67μs
	Recall Tree	11.4%	8.6%	1972s	194μs
	LOMTree	21.4%	16.5%*	112s	28μs
Imagenet	OAA	84.7%	82.2%	446d (20.4h)	118ms
	Recall Tree	91.1%	88.4%	71.4h	4ms
	LOMTree	96.7%	90.1%*	14.0h	0.56ms
LTCB	OAA	78.7%	76.8%	764d (19.1h)	3600μs
	Recall Tree	78.0%	77.6%	4.8h	76μs
	LOMTree	78.4%	-	4.3h	51μs
ODP	OAA	91.2%	90.6%	133d (1.3h)	560ms
	Recall Tree	96.2%	93.8%	1.5h	1.9ms
	LOMTree	95.4%	93.5%*	0.6h	0.52ms

for the smaller data set (ALOI), we do multiple passes over the training data, monitoring a 10% held-out portion of the training data to determine when to stop optimizing.

Tuned Performance Methodology For tuned performance, we use random search over hyperparameters, taking the best result over 59 probes. For the smaller data set (ALOI), we optimize validation error on a 10% held-out subset of the training data. For the larger data sets (Imagenet, ODP), we optimize progressive validation loss on the initial 10% of the training data. After determining hyperparameters we retrain with the entire training set and report the resulting test error.

When available we report published LOMtree results, although they utilized a different method for optimizing hyperparameters.

Timing Measurements All timings are taken from the same 24 core xeon server machine. Furthermore, all algorithms are implemented in the Vowpal Wabbit toolkit and therefore share file formats, parser, and binary classification base learner implying differences are attributable to the different reductions. Our baseline OAA implementation is mature and highly tuned: it always exploits vectorization, and furthermore can optionally utilize multicore training and negative gradient subsampling to accelerate training. For the larger datasets these latter features were necessary to complete the experiments: estimated unaccelerated training times are given, along with wall clock times in parenthesis.