Improved Multiclass AdaBoost for Image Classification: the Role of Tree Optimization

Arman Zharmagambetov, **Magzhan Gabidolla** and **Miguel Á. Carreira-Perpiñán**

Dept. Computer Science & Engineering, UC Merced

Poster Number: 3001

1 Abstract

Decision tree boosting is considered as an important and widely recognized method in image classification, despite dominance of the deep learning based approaches in this area. Provided with good image features, it can produce a powerful model with unique properties, such as strong predictive power, scalability, interpretability, etc. In this paper, we propose a novel tree boosting framework which capitalizes on the idea of using shallow, sparse and yet powerful oblique decision trees (trained with recently proposed *Tree Alternating Optimization* algorithm) as the base learners. We empirically show that the resulting model achieves better or comparable performance (both in terms of accuracy and model size) against established boosting algorithms such as gradient boosting or AdaBoost in number of benchmarks. Further, we show that such trees can directly and efficiently handle multiclass problems without using one-vs-all strategy employed by most of the practical boosting implementations.

Work supported by NSF award IIS–2007147

2 AdaBoost and Decision Trees

Ensembles of decision trees have numerous applications in many tasks in image processing and computer vision. Among others, AdaBoost with decision tree base learners is often referred to as the best off-the-shelf classifier for various problems.

- Many papers and current implementations of AdaBoost use trees that are:
  - Axis-aligned (i.e. it uses a single feature at a decision node)
  - Trained with greedy top-down induction
- However, axis-aligned trees are not very suitable for many problems, especially for the ones with correlated features (e.g. pixels of an image).
- Greedy top-down induction produces suboptimal trees.
- Because of these, boosting algorithms usually need to induce K (= number of classes) such trees at each boosting step, which adds an extra overhead.
- And, to find a suitable splitting criterion for specific objective functions (as is the case with many boosting algorithms) is not straightforward with these greedy algorithms.

3 Optimizing trees in AdaBoost.MH with TAO

Our proposal:

- to use oblique decision trees (i.e. trees with hyperplane splits at decision nodes)
- to use a non-greedy optimization algorithm to learn such trees

We adapt the recently proposed algorithm for learning decision trees, Tree Alternating Optimization (TAO) for AdaBoost.MH and empirically evaluate its performance on image classification datasets.

Given a tree structure $\mathbf{T}$, a training set $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$, and AdaBoost.MH weights $\{w_{n,k}\}_{n=1,k=1}^{N,K}$, TAO directly optimizes the following base learner's objective function with a guarantee of monotonically decreasing it:

$$E(\mathbf{\Theta}) = \sum_{n=1}^{N} \sum_{k=1}^{K} w_{n,k} \cdot \exp(-y_{n,k} \cdot \mathbf{T}_k(\mathbf{x}_n)) \qquad (1)$$

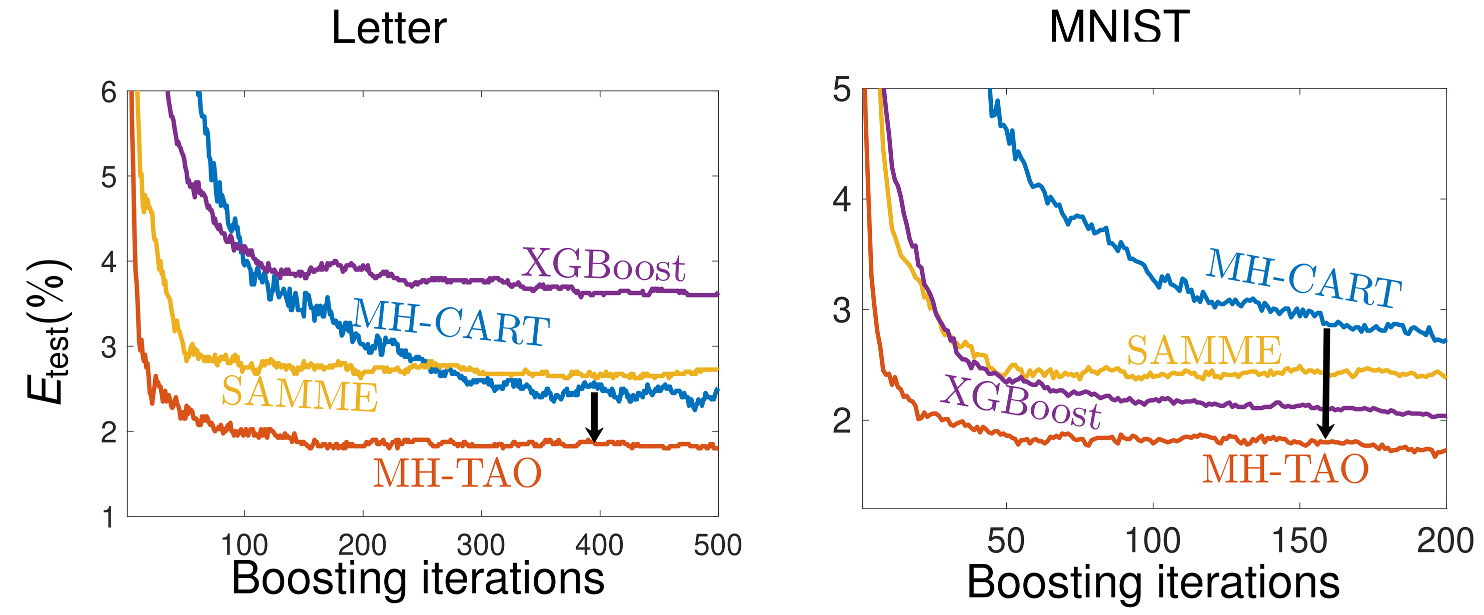
Previous implementations of AdaBoost.MH used K (= number of classes) trees at each boosting step. In this work we only fit a single oblique decision tree trained with TAO.

Algorithm 1: Learning a tree for AdaBoost.MH with TAO

input training set $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$;
initial tree $\mathbf{T}(\cdot; \mathbf{\Theta})$ of depth Δ;
AdaBoost.MH weights $\{w_{n,k}\}_{n=1,k=1}^{N,K}$;
repeat
  for *depth* $d = 0$ *to* Δ **do**
    for $i \in$ *nodes at depth* d **do**
      if i *is a leaf* **then**
        $\mathbf{y}_i \leftarrow$ find a constant vector to minimize eq. 1 (has a closed form solution);
      else
        $\boldsymbol{\theta}_i \leftarrow$ fit a weighted binary classifier;
      end
    end
  end
until *convergence occurs or max iteration*;
return *trained tree* $\mathbf{T}$

4 Experiments

We experimentally evaluate AdaBoost.MH with TAO trees against established forest-based methods. Boosted TAO trees are smaller (fewer and shallower trees) yet consistently more accurate.



	Forest	E_{test} (%)	#pars.	FLOPS	T	Δ
MNIST	RF	3.05±0.06	1M	(3 482)	100	46
	SAMME	2.96±0.05	6M	(29 489)	1 000	30
	RF	2.84±0.06	10M	(34 507)	1 000	48
	sNDF	2.80±0.12	22M	(22M)	80	10
	XGBoost	2.73±0.00	390k	(16 812)	1 000	30
	MH-CART	2.73±0.00	307k	(1 400)	200	7
	ADF	2.71±0.10	3.6M	(2 500)	100	25
	XGBoost	2.67±0.00	324k	(8 000)	1 000	8
	SAMME	2.28±0.02	13.3M	(16 000)	1 000	16
	XGBoost	2.17±0.00	540k	(57 385)	10 000	30
	rRF	2.05±0.02	(160k)	(2 500)	100	25
	MH TAO	1.96±0.06	837k	54 041	20	8
	XGBoost	1.94±0.00	615k	(51 873)	10000	8
	MH-TAO	1.92±0.07	2.3M	93 523	30	8
	MH-TAO	1.72±0.08	7.9M	312 904	100	8
ImageNet subset	MH-CART	>8 days runtime			100	9
	MH-CART	25.07	9400	(500)	100	5
	RF	13.62±0.32	2.5M	(23k)	100	220
	RF	12.67±0.13	12.7M	(109k)	500	218
	RF	12.51±0.11	25.4M	(224k)	1000	220
	XGBoost	12.51±0.00	596k	(81k)	6400	50
	XGBoost	11.01±0.00	782k	(124k)	32000	50
	XGBoost	10.78±0.00	973k	(181k)	64000	50
	MH-TAO	10.65±0.05	3.4M	105k	30	12