Improved Algorithms for Distributed Boosting

Jeff Cooper
Google

Lev Reyzin
University of Illinois at Chicago

What is Boosting?

weak learning
achieve some error
ε < ½ on all
distributions

[Schapire ’90]

boosting

strong learning
achieve any error ε > 0
on all distributions

AdaBoost

Given: $(x_1, y_1), \ldots, (x_n, y_n)$
where $x_i \in X$, $y_i \in Y = \{-1, +1\}$.
Initialize $D_1(i) = 1/m$.
for $t = 1, \ldots, T$ **do**
Train base learner using distribution D_t.
Get base classifier $h_t : X \to \{-1, +1\}$.
Let $\gamma_t = \sum_i D_t(i) y_i h_t(x_i)$.
Choose $\alpha_t = \frac{1}{2} \ln \frac{1+\gamma_t}{1-\gamma_t}$.
Update:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t},$$

where Z_t normalizes so that D_{t+1} is a distribution.
end for
Output the final classifier:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right).$$

Distributed Boosting

What if data does not all fit on one machine? How can we distribute Boosting with generic weak learners?

Learning Theory

Here, our goal is <u>not to simulate AdaBoost</u> as efficiently as possible, but rather to create **practical algorithms**.

Distributed boosting has been studied in the PAC and agnostic settings, especially considering the communication complexity of the resulting algorithms [Balcan et al., 2012; Chen et al., 2016]. We leave comparison to these methods for future work.

“The Distributed Boosting Algorithm”

[Lazarevic-Obradovic ’01] proposed an algorithm to do “boosting”:

- Data is partitioned among machines
- Each machine keeps local distribution
- Each trains weak learner and “majority rule” is used.
- Concatenation of local distributions mimics global distribution
- All machines communicate pairwise each round.

DistBoost

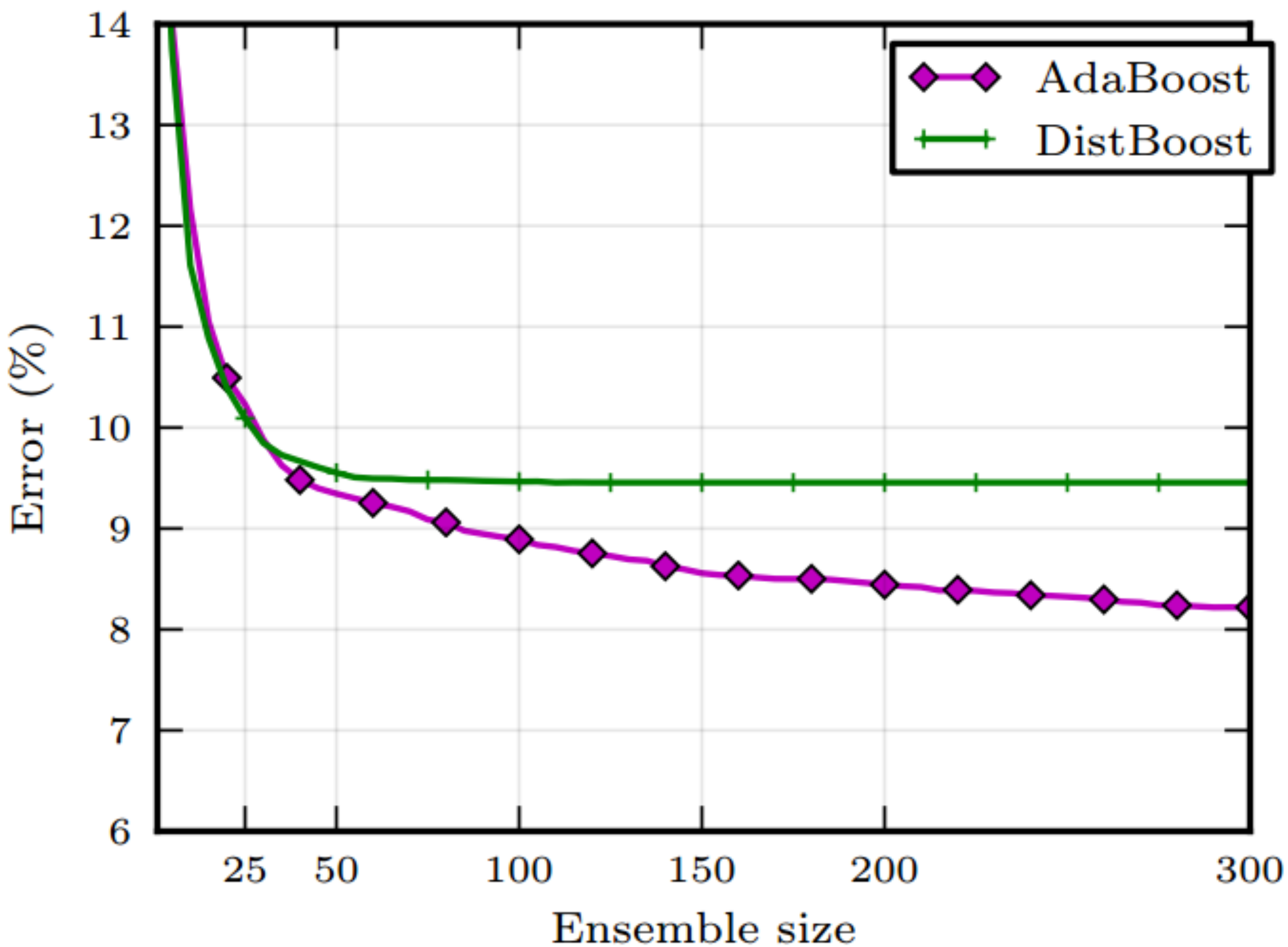
Given: K machines, $(x_1, y_1), \ldots, (x_{Kn}, y_{Kn})$
where $x_i \in X$, $y_i \in Y = \{-1, +1\}$.
Initialize $D_1(i) = \frac{1}{Kn}$.
for $t = 1, \ldots, T$ **do**
  for $j = 1, \ldots, K$ (in parallel) **do**
    Train base learner using data at site j and dist. D_t.
    Get base classifier $h_{t,j} : X \to \{-1, +1\}$.
  end for
  Let $E_t(x) = \text{sign}\left(\sum_{j=1}^{K} h_{t,j}(x)\right)$.
  Let $\gamma_t = \sum_i D_t(i) y_i E_t(x_i)$.
  Choose $\alpha_t = \frac{1}{2} \ln \frac{1+\gamma_t}{1-\gamma_t}$.
  Update:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i E_t(x_i))}{Z_t},$$

  where Z_t normalizes so that D_{t+1} is a distribution.
end for
Output the final classifier:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

DistBoost Overfits



← On UCI particle dataset

Two problems

- No “local” weak learner is good on global distribution.
- Too much communication.

DIvoting: A Proposed Improvement

- **DIVoting** [Chawla et al., 2004] distributes Breiman's Ivoting method ['96].

- **Ivoting** creates a classifier by resampling the data (like Bagging) so that current ensemble keeps getting about half of the dataset correct.

- Final distributed classifier just combines all ensembles into one **large majority vote**.

- This creates a distributed classifier with **little communication**. But, it's **not a boosting method** and doesn't drive down training error.

Two Proposed Solutions

- **PreWeak:** preselect good weak learners and updates based on global distribution. (downside: lots of communication)

- **AdaSampling**: uses boosting to send informative examples to one machine (downside: discards examples)

PreWeak

Given: K machines, $(x_1, y_1), \ldots, (x_{Kn}, y_{Kn})$
where $x_i \in X$, $y_i \in Y = \{-1, +1\}$.
for $j = 1, \ldots, K$ (in parallel) **do**
   Run `AdaBoost` for T rounds using data at site j
   Save collection of weak learners $h_{j,1} \ldots, h_{j,T}$.
end for
Initialize $D_1(i) = \frac{1}{Kn}$.
for $t = 1, \ldots, T$ **do**
   Choose h_t from collection

$$\{h_{j,i} : 1 \le j \le K, 1 \le i \le T\}$$

   that minimizes error with respect to D_t.
   Let $\gamma_t = \sum_i D_t(i) y_i h_t(x_i)$.
   Choose $\alpha_t = \frac{1}{2} \ln \frac{1+\gamma_t}{1-\gamma_t}$.
   Update:

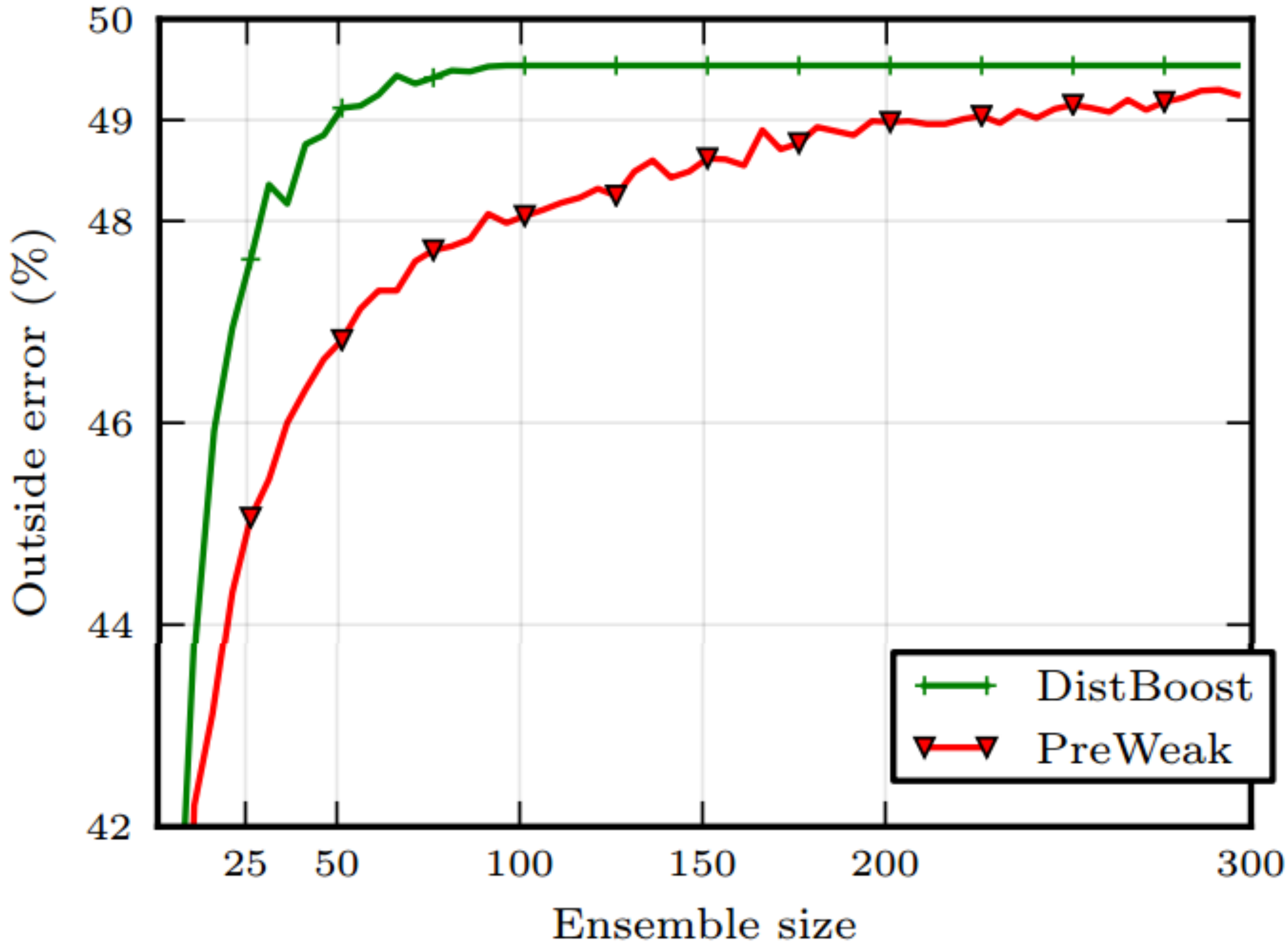
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t},$$

   where Z_t normalizes so that D_{t+1} is a distribution.
end for
Output the final classifier:

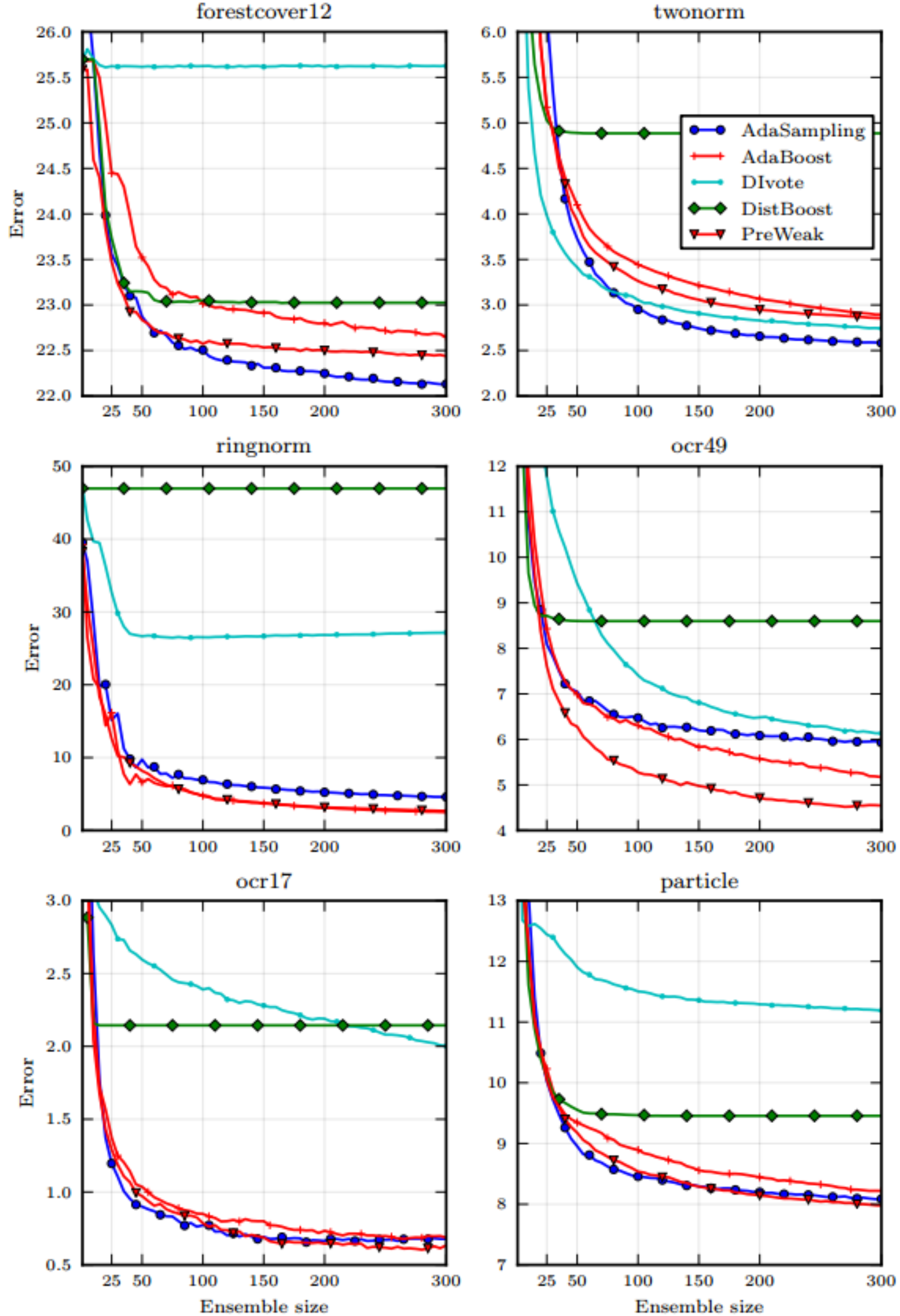
$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

Adaptive Sampling

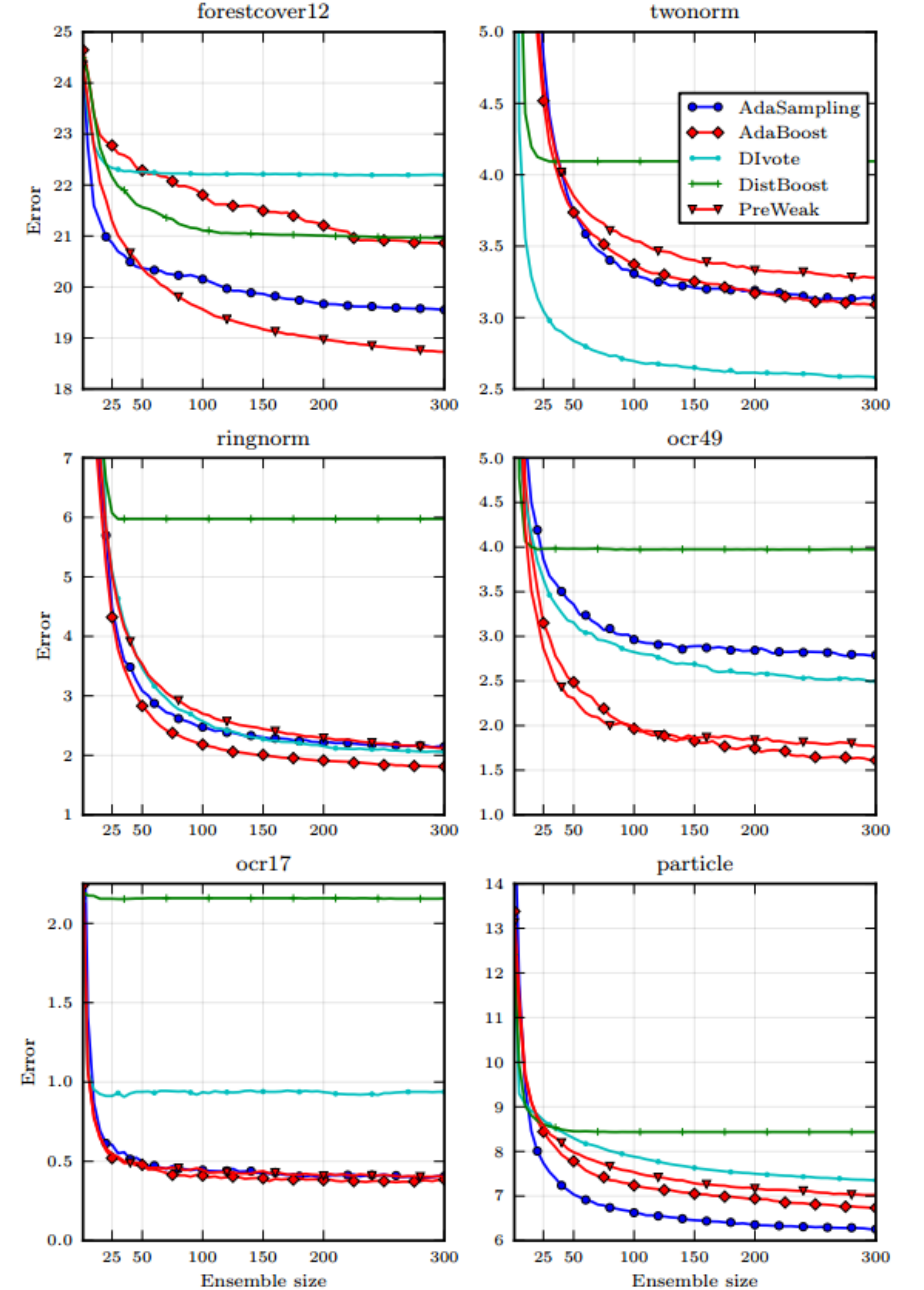
Given: K machines, $(x_1, y_1), \ldots, (x_{Kn}, y_{Kn})$
where $x_i \in X$, $y_i \in Y = \{-1, +1\}$.
for $j = 1, \ldots, K$ (in parallel) **do**
  Run `AdaBoost` for T rounds using data at site j
  Sort examples by decreasing value of $\sum_{t=1}^{T} D_t^j(i)/t$
  Broadcast n/K consecutive examples with lowest local test error
end for
Run `AdaBoost` with training set of the n broadcasted examples.
Output classifier returned by `AdaBoost`



Outside error (%)
50
49
48
46
44
42
25
50
100
150
200
300
Ensemble size
DistBoost
PreWeak



stumps



depth-3 trees

Discussion

- We presented two new algorithms for distributed boosting.
- Both of our algorithms are **competitive with AdaBoost** when it is trained with the entire dataset. Both algorithms outperform DistBoost in all our experiments and Devoting in most experiments.
- **PreWeak** was able to boost its accuracy at the same rate as AdaBoost.
- **AdaSampling** (like DIvoting) requires no communication between sites yet outperformed it on several datasets. AdaSampling, however, was substantially worse than AdaBoost on two of the datasets.
- It remains open to create a boosting algorithm that is always competitive with AdaBoost yet requires as little communication as DIvote.