Aalto University
School of Science

CS-E4070 — Computational learning theory

Slide set 06 : AdaBoost

Cigdem Aslay and Aris Gionis

Aalto University

spring 2019

summary

- a weak learner can be transformed to a strong learner
- confidence can be boosted by iterative runs of the weak learner
- accuracy can be boosted by focusing on regions of the target distribution that are more difficult to learn
  - two step process:
    1. reduce error quadratically
    2. recursive application of 1. to reduce error to ϵ
- analysis is quite involved (in particular the recursive part)
- algorithm is not practical
- can we design a practical boosting algorithm?
  yes! AdaBoost

reading material

- Schapire, “The boosting approach to machine learning: an overview”, 2001
- Freund and Schapire, “Boosting: foundations and algorithms”. MIT press, 2012
  available as an e-book by the Aalto library services

the AdaBoost algorithm

high-level idea

- start with a training set
- assume that we can train a weak learner
- apply the weak learner repeatedly
  - each time with a different weighting scheme of the training data
- in each iteration learn a different hypothesis
- combine those hypotheses into a single hypothesis

considerations

1. how to choose the weighting schemes for each iteration?
2. how to combine the learned hypotheses?

the AdaBoost algorithm

input: training set $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$,
where $\mathbf{x}_i \in X$ and $y_i \in Y = \{-1, +1\}$

initialize $\mathcal{D}_1(i) = 1/m$, for all $i = 1, \ldots, m$

for $t = 1, \ldots, T$

train weak learner A_w using $\mathcal{D}_t$

find hypothesis h_t

set parameter α_t according to the accuracy of h_t on $\mathcal{D}_t$

update $\mathcal{D}_{t+1}(i) = \frac{1}{Z_t}\mathcal{D}_t(i)e^{-\alpha_t y_i h_t(\mathbf{x}_i)}$, for all $i = 1, \ldots, m$

(where Z_t is a normalization parameter)

return $h = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t\right)$

the AdaBoost algorithm

- the weak learner A_w at round t aims to minimize the error

$$\epsilon_t = \mathbf{Pr}_{\mathbf{x}_i \sim \mathcal{D}_t}[h_t(\mathbf{x}_i) \neq y_i]$$

- in the binary case ($Y = \{-1, +1\}$) we typically set

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

the training error of AdaBoost

- it can be shown that AdaBoost is able to reduce its error on the training set

analysis (sketch)

- define $f(\mathbf{x}) = \sum_t \alpha_t h_t(\mathbf{x})$, so $h(\mathbf{x}) = \text{sign}(f(\mathbf{x}))$

  if $h(\mathbf{x}_i) \neq y_i$ then $1 \leq e^{-y_i f(\mathbf{x}_i)}$

- the training error is

$$\frac{1}{m} |i : h(\mathbf{x}_i) \neq y_i| \leq \frac{1}{m} \sum_i e^{-y_i f(\mathbf{x}_i)} = \prod_t Z_t$$

  the equality is shown using the recursive definition of $\mathcal{D}_t$

the training error of AdaBoost (analysis, cont'd)

- to minimize the training error we want to choose α_t and h_t to minimize

$$Z_t = \sum_i \mathcal{D}_t(i) e^{-\alpha_t y_i h_t(i)}$$

in each round

- for the choice of $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ we get

$$\prod_t Z_t \leq \prod_t 2\sqrt{\epsilon_t(1-\epsilon_t)} = \prod_t \sqrt{1-4\gamma_t^2} \leq e^{-2\sum_t \gamma_t^2}$$

where $\gamma_t = \frac{1}{2} - \epsilon_t$

the training error of AdaBoost (analysis, cont'd)

- assume that the weak learner A is better than random in each round
- then $\epsilon_t < \frac{1}{2}$, and so $\gamma_t = \frac{1}{2} - \epsilon_t \geq \gamma$, for some $\gamma > 0$
- thus, the training error is bounded by

$$\frac{1}{m} |i : h(\mathbf{x}_i) \neq y_i| \leq e^{-2\sum_t \gamma_t^2} \leq e^{-2T\gamma^2}$$

- we conclude that AdaBoost is a true boosting algorithm, where the error drops exponentially fast in T
- however, this is training error

some intuition on AdaBoost

- we seek to minimize the error

$$\sum_i e^{-y_i f(\mathbf{x}_i)} = \sum_i e^{-y_i \sum_t \alpha_t h_t(\mathbf{x}_i)}$$

  where in each iteration we choose a new hypothesis h_t and coefficient α_t

- can be seen as steepest-descend optimization where the search is constrained to follow coordinate directions
  - base classifiers h_t define the coordinates

the generalization error of AdaBoost

- however, we are primarily interested to bound the generalization error
- Freund and Schapire showed how to obtain generalization error bounds for AdaBoost
- assume m samples, VC dimension d, and T rounds
- generalization error bound of AdaBoost was shown to be

$$\hat{\mathbf{Pr}}[h(\mathbf{x}_i) \neq y_i] + \tilde{\mathcal{O}}\left(\sqrt{\frac{Td}{m}}\right)$$

  where $\hat{\mathbf{Pr}}[\cdot]$ denotes empirical probability on the training set and $\tilde{\mathcal{O}}(\cdot)$ *"hides"* polylogarithmic factors
- factor T in the numerator suggests overfitting

the generalization error of AdaBoost

- in practice the generalization error bound of AdaBoost is often better than what theory suggests

$$\hat{\mathbf{Pr}}[h(\mathbf{x}_i) \neq y_i] + \tilde{\mathcal{O}}\left(\sqrt{\frac{Td}{m}}\right)$$

- in practice AdaBoost often does not overfit
  - while bound suggests overfitting with T
- in practice generalization error decreases long after training error has reached to 0

generalization error in terms of margins

- the margin of an example $(\mathbf{x}, y)$ is defined as

$$\text{margin}_f(\mathbf{x}, y) = \frac{y f(\mathbf{x})}{\sum_t |\alpha_t|} = \frac{y \sum_t \alpha_t h_t(\mathbf{x})}{\sum_t |\alpha_t|}$$

- margin is a number in $[-1, +1]$
- positive margin indicates that the example is classified correctly
- magnitude of margin can be interpreted as a measure of confidence in the prediction of the classifier

generalization error in terms of margins

- the larger the margin on the training set, the smaller the generalization error
- Schapire et al. showed that generalization error is

$$\hat{\mathbf{Pr}}[\text{margin}_f(\mathbf{x}, y) \leq \theta] + \tilde{\mathcal{O}}\left(\sqrt{\frac{d}{m\theta^2}}\right)$$

- independent on the number of rounds T
- margins are useful to understand the behavior of boosting in practice
- e.g., in practice margin in the training set may keep increasing even after training error has reached 0

in practice

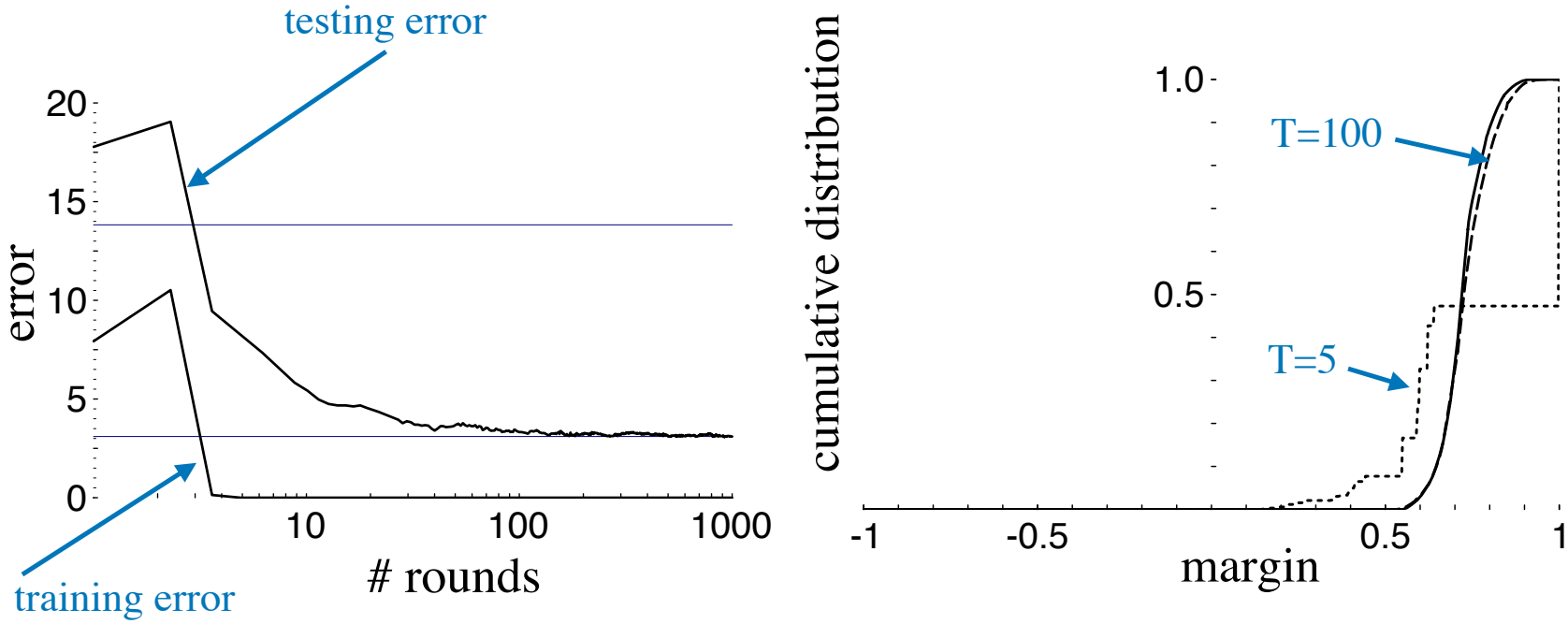

Figure 2: Error curves and the margin distribution graph for boosting C4.5 on the letter dataset as reported by Schapire et al. [69]. *Left*: the training and test error curves (lower and upper curves, respectively) of the combined classifier as a function of the number of rounds of boosting. The horizontal lines indicate the test error rate of the base classifier as well as the test error of the final combined classifier. *Right*: The cumulative distribution of margins of the training examples after 5, 100 and 1000 iterations, indicated by short-dashed, long-dashed (mostly hidden) and solid curves, respectively.

summary

- AdaBoost is an important learning algorithm
- rigorous theoretical analysis, and works well in practice
- many extensions, interpretations, connections
  - e.g., extensions to multi-class classification, logistic regression
  - connections to stochastic optimization, linear programming, game theory