Boosting

Machine Learning: Jordan Boyd-Graber
University of Maryland
SLIDES ADAPTED FROM ROB SCHAPIRE

Motivating Example

Goal

We have a bunch of classifiers; how do we get best possible combination?

- SVM works well on X
- Neural model works well on Y
- Logistic Regression works well on Z
- Hard to know which model to use!

Motivating Example

Goal

We have a bunch of classifiers; how do we get best possible combination?

- SVM works well on X
- Neural model works well on Y
- Logistic Regression works well on Z
- Hard to know which model to use!
- Most Kaggle competitions won using ensemble approaches

Boosting Approach

- devise computer program for deriving rough rules of thumb
- apply procedure to subset of examples
- obtain rule of thumb
- apply to second subset of examples
- obtain second rule of thumb
- repeat T times

Details

- How to **choose** examples
- How to **combine** rules of thumb

Details

- How to **choose** examples
  concentrate on hardest examples (those most often misclassified by previous rules of thumb)
- How to **combine** rules of thumb

Details

- How to **choose** examples
  concentrate on hardest examples (those most often misclassified by previous rules of thumb)
- How to **combine** rules of thumb
  take (weighted) majority vote of rules of thumb

Boosting

Definition

general method of converting rough rules of thumb into highly accurate prediction rule

- assume given weak learning algorithm that can consistently find classifiers (rules of thumb) at least slightly better than random, say, accuracy $\geq 55\%$ (in two-class setting)
- given sufficient data, a boosting algorithm can provably construct single classifier with very high accuracy, say, 99%

Formal Description

- Training set $(x_1, y_1) \dots (x_m, y_m)$
- $y_i \in \{-1, +1\}$ is the label of instance x_i

Formal Description

- Training set $(x_1, y_1) \ldots (x_m, y_m)$
- $y_i \in \{-1, +1\}$ is the label of instance x_i
- For $t = 1, \ldots T$:
    - Construct distribution D_t on $\{1, \ldots, m\}$
    - Find weak classifier

$$h_t : \mathscr{X} \mapsto \{-1, +1\} \tag{1}$$

with small error ϵ_t on D_t:

$$\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i] \tag{2}$$

Formal Description

- Training set $(x_1, y_1) \dots (x_m, y_m)$
- $y_i \in \{-1, +1\}$ is the label of instance x_i
- For $t = 1, \dots T$:
    - Construct distribution D_t on $\{1, \dots, m\}$
    - Find weak classifier

$$h_t : \mathscr{X} \mapsto \{-1, +1\} \tag{1}$$

    with small error ϵ_t on D_t:

$$\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i] \tag{2}$$

    - Output final classifier H_{final}

AdaBoost (Schapire and Freund)

- Data distribution D_t

AdaBoost (Schapire and Freund)

- Data distribution D_t
  - $D_1(i) = \frac{1}{m}$
  - Given D_t and h_t:

$$D_{t+1}(i) \propto D_t(i) \cdot \exp\left\{-\alpha_t y_i h_t(x_i)\right\} \tag{3}$$

where $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) > 0$

AdaBoost (Schapire and Freund)

- Data distribution D_t
  - $D_1(i) = \frac{1}{m}$
  - Given D_t and h_t:

$$D_{t+1}(i) \propto D_t(i) \cdot \exp\left\{-\alpha_t y_i h_t(x_i)\right\} \tag{3}$$

where $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) > 0$

Bigger if wrong, smaller if right

AdaBoost (Schapire and Freund)

- Data distribution D_t
  - $D_1(i) = \frac{1}{m}$
  - Given D_t and h_t:

$$D_{t+1}(i) \propto D_t(i) \cdot \exp\left\{-\alpha_t y_i h_t(x_i)\right\} \quad (3)$$

where $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) > 0$

Weight by how good the weak learner is

AdaBoost (Schapire and Freund)

- Data distribution D_t
  - $D_1(i) = \frac{1}{m}$
  - Given D_t and h_t:

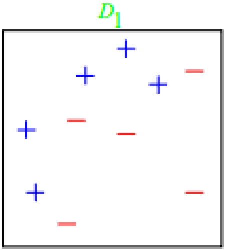
$$D_{t+1}(i) \propto D_t(i) \cdot \exp\left\{-\alpha_t y_i h_t(x_i)\right\} \tag{3}$$

  where $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) > 0$

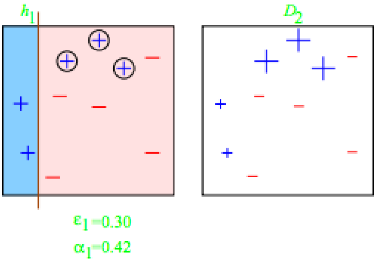
- Final classifier:

$$H_{fin}(x) = \text{sign}\left(\sum_t \alpha_t h_t(x)\right) \tag{4}$$

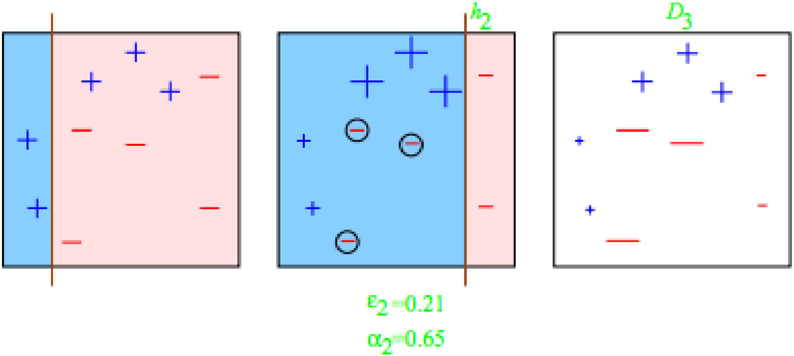
Toy Example



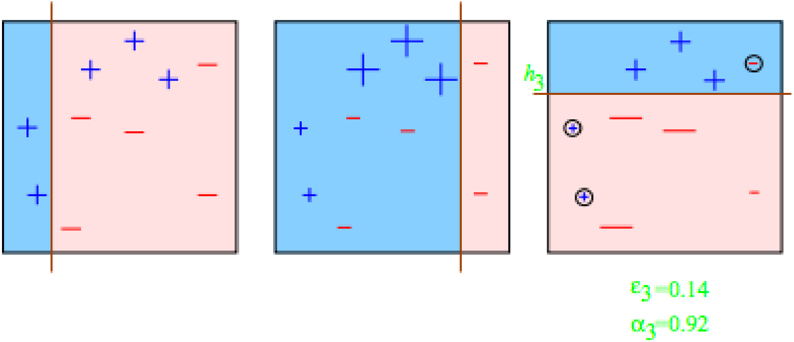
Round 1



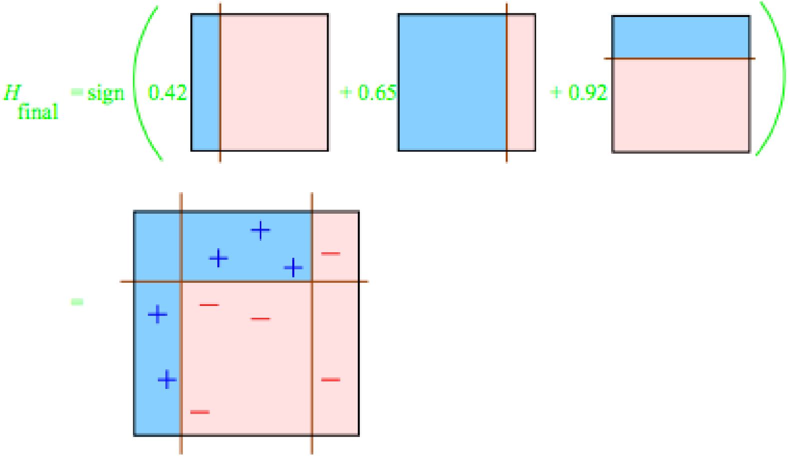
Round 2



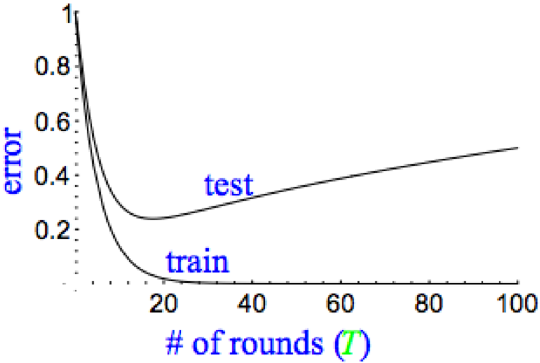
Round 3



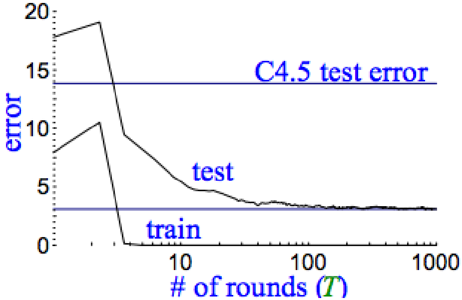
Final Classifier



Generalization



Generalization



(boosting C4.5 on "letter" dataset)

Practical Advantages of AdaBoost

- fast
- simple and easy to program
- no parameters to tune (except T)
- flexible: can combine with any learning algorithm
- no prior knowledge needed about weak learner
- provably effective, provided can consistently find rough rules of thumb
  - shift in mind set: goal now is merely to find classifiers barely better than random guessing
- versatile
  - can use with data that is textual, numeric, discrete, etc.
  - has been extended to learning problems well beyond binary classification

Caveats

- performance of AdaBoost depends on data and weak learner
- consistent with theory, AdaBoost can fail if
- weak classifiers too complex
  - overfitting
- weak classifiers too weak ($\gamma_t \longrightarrow 0$ too quickly)
  - underfitting
  - low margins $\longrightarrow$ overfitting
- empirically, AdaBoost seems especially susceptible to uniform noise