Department of Computer Science
UNIVERSITY OF COLORADO **BOULDER**

Boosting

Jordan Boyd-Graber
University of Colorado Boulder
LECTURE 12

Slides adapted from Rob Schapire

Motivating Example

Goal

Automatically categorize type of call requested by phone customer (Collect, CallingCard, PersonToPerson, etc.)

- yes I'd like to place a collect call long distance please (Collect)
- operator I need to make a call but I need to bill it to my office (ThirdNumber)
- yes I'd like to place a call on my master card please (CallingCard)
- I just called a number in sioux city and I musta rang the wrong number because I got the wrong party and I would like to have that taken off of my bill (BillingCredit)

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%

Plan

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

$$h_t : \mathcal{X} \mapsto \{-1, +1\} \quad (1)$$

  with small error ϵ_t on D_t:

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

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

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

    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:

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

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:

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

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:

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

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:

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

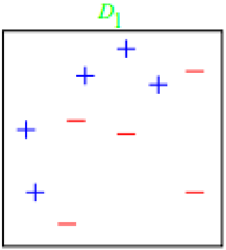
  where $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right) > 0$

- Final classifier:

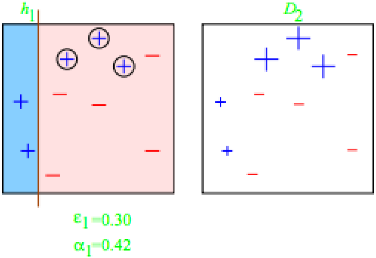
$$H_{fin}(x) = \operatorname{sign}\left(\sum_t \alpha_t h_t(x) \right) \tag{4}$$

Plan

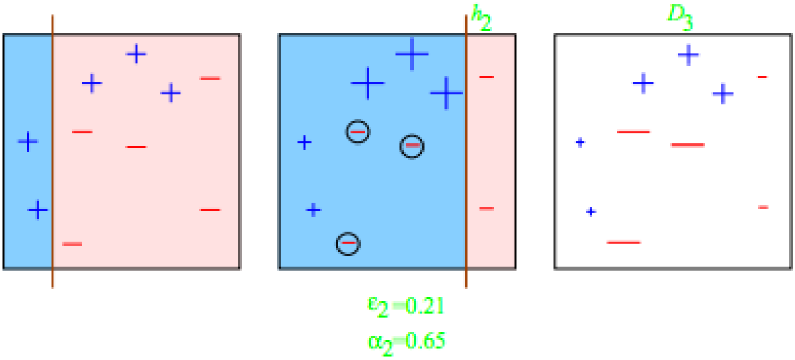
Toy Example



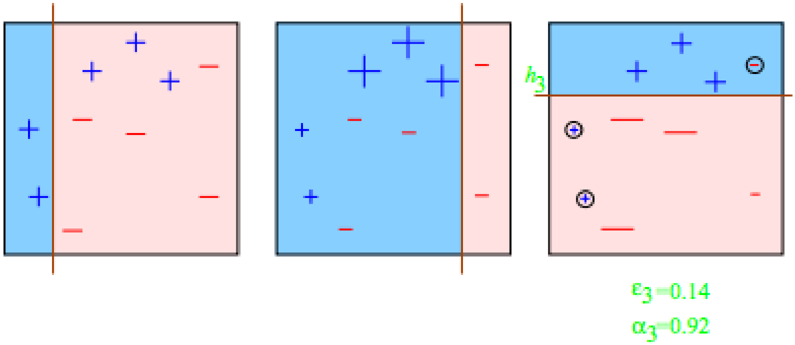
Round 1



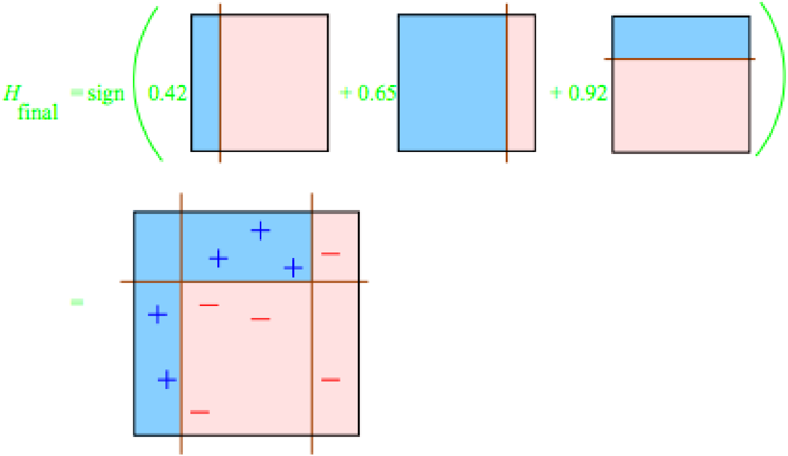
Round 2



Round 3



Final Classifier



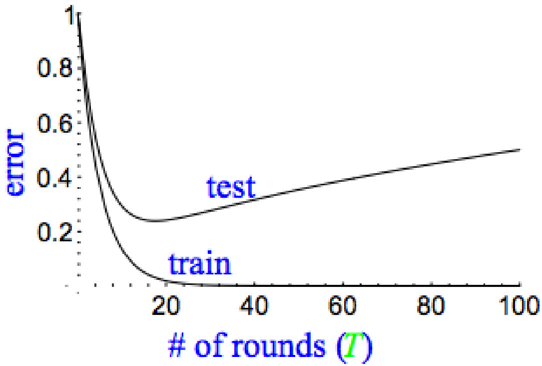
Plan

Algorithm

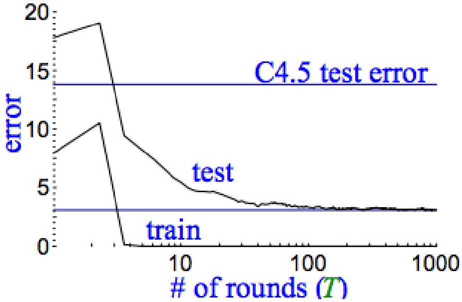
Example

Generalization

Generalization



Generalization



(boosting C4.5 on "letter" dataset)

Theoretical Analysis

- Training error $(\epsilon_t = \frac{1}{2} - \gamma_t)$

$$\leq \exp\left\{-2\sum_t \gamma_t^2\right\} \tag{5}$$

- Generalization error

$$\leq \Pr\left[\text{margin} \leq \theta\right] + \mathcal{O}\left(\frac{\sqrt{d/m}}{\theta}\right) \tag{6}$$

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 ($\gamma_t \rightarrow 0$ too quickly)
    - underfitting
    - low margins $\rightarrow$ overfitting
- empirically, AdaBoost seems especially susceptible to uniform noise

Next Time

- Online learning
- Dealing with chafing data
- How to deal with error in this case