Boosting, HW Help

Colin White, Kenny Marino,
Nupur Chatterji

Plan for Today

- Review Boosting, Adaboost
- HW4 questions

Slides by Rob Schapire, Nina Balcan, Colin White

Boosting Idea

- devise computer program for deriving rough rules of thumb
- apply procedure to subset of examples
- obtain rule of thumb
- apply to 2nd subset of examples
- obtain 2nd rule of thumb
- repeat T times

Boosting Idea

- how to choose examples on each round?
    - concentrate on "hardest" examples (those most often misclassified by previous rules of thumb)
- how to combine rules of thumb into single prediction rule?
    - take (weighted) majority vote of rules of thumb

Boosting Idea

- boosting = general method of converting rough rules of thumb into highly accurate prediction rule
- technically:
  - 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%

Strong and Weak Learnability

- **Strong PAC learning algorithm:**
    - Learns classifier with error 1%, with high probability, for any distribution
- **Weak PAC learning algorithm:**
    - Learns classifier with error 49%, with high probability, for any distribution

Boosting: weak learning implies strong learning

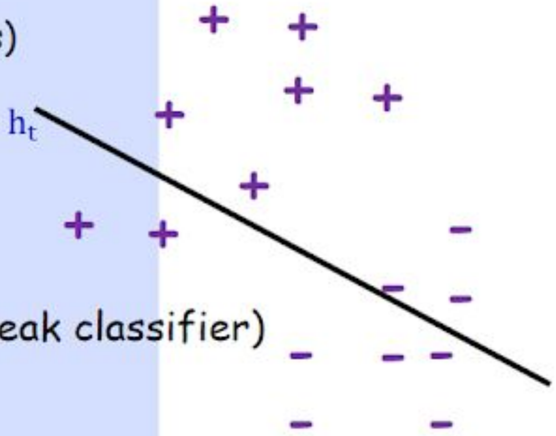
Adaboost

Input: $S=\{(x_1,y_1), \ldots, (x_m,y_m)\}$; $x_i \in X, y_i \in Y = \{-1,1\}$

weak learning algo A (e.g., Naïve Bayes, decision stumps)

- For $t=1,2, \ldots, T$
  - Construct D_t on $\{x_1, \ldots, x_m\}$
  - Run A on D_t producing $h_t: X \rightarrow \{-1,1\}$ (weak classifier)

    $\epsilon_t = P_{x_i \sim D_t}(h_t(x_i) \neq y_i)$ error of h_t over D_t
- Output $H_{final}(x) = \text{sign}(\sum_{t=1} \alpha_t h_t(x))$



Adaboost

- Weak learning algorithm *A*.
- For t=1,2, … ,T
  - **Construct D_t on $\{x_1, \ldots, x_m\}$**
  - Run *A* on D_t producing h_t

Constructing D_t

- D_1 uniform on $\{x_1, \ldots, x_m\}$ [i.e., $D_1(i) = \frac{1}{m}$]
- Given D_t and h_t set

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} e^{\{-\alpha_t\}} \text{ if } y_i = h_t(x_i)$$

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} e^{\{\alpha_t\}} \text{ if } y_i \neq h_t(x_i)$$

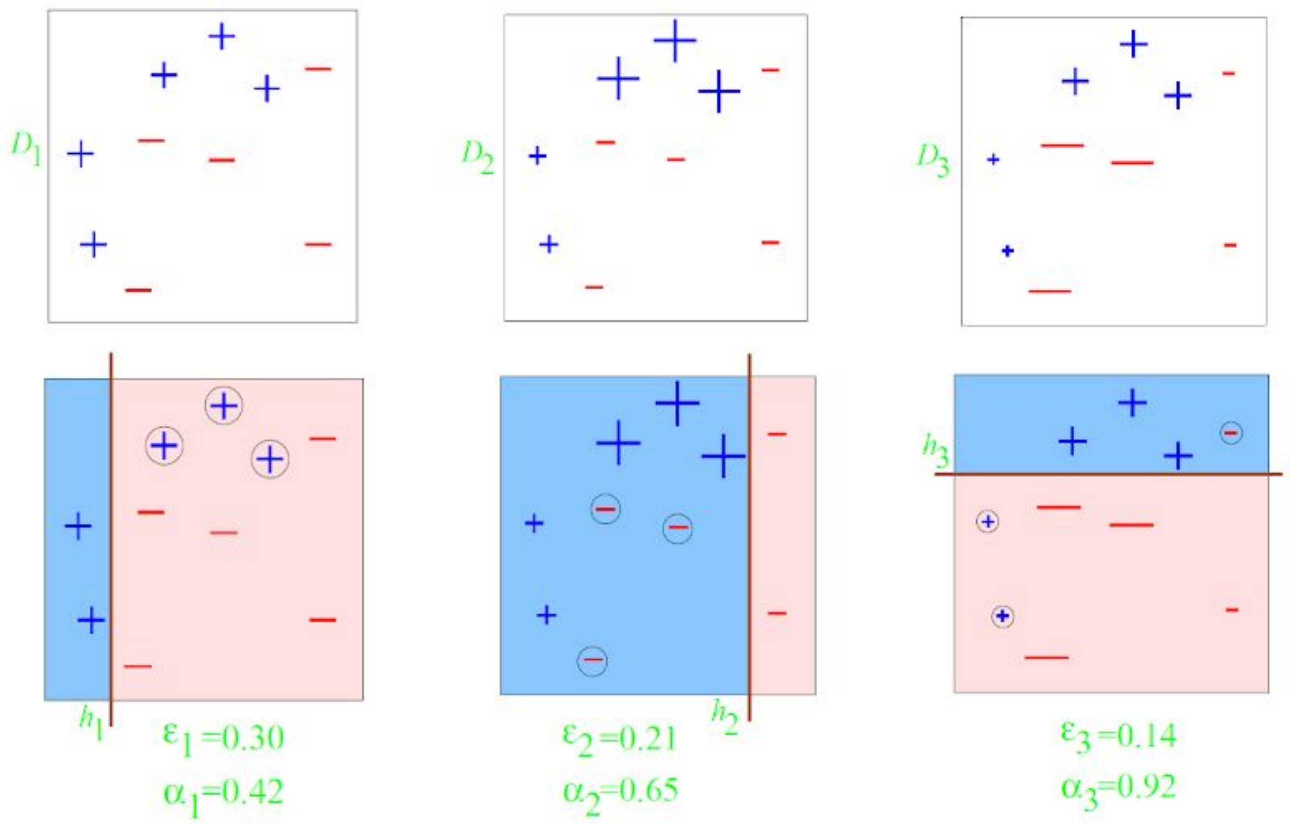
$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} e^{\{-\alpha_t y_i h_t(x_i)\}}$$

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

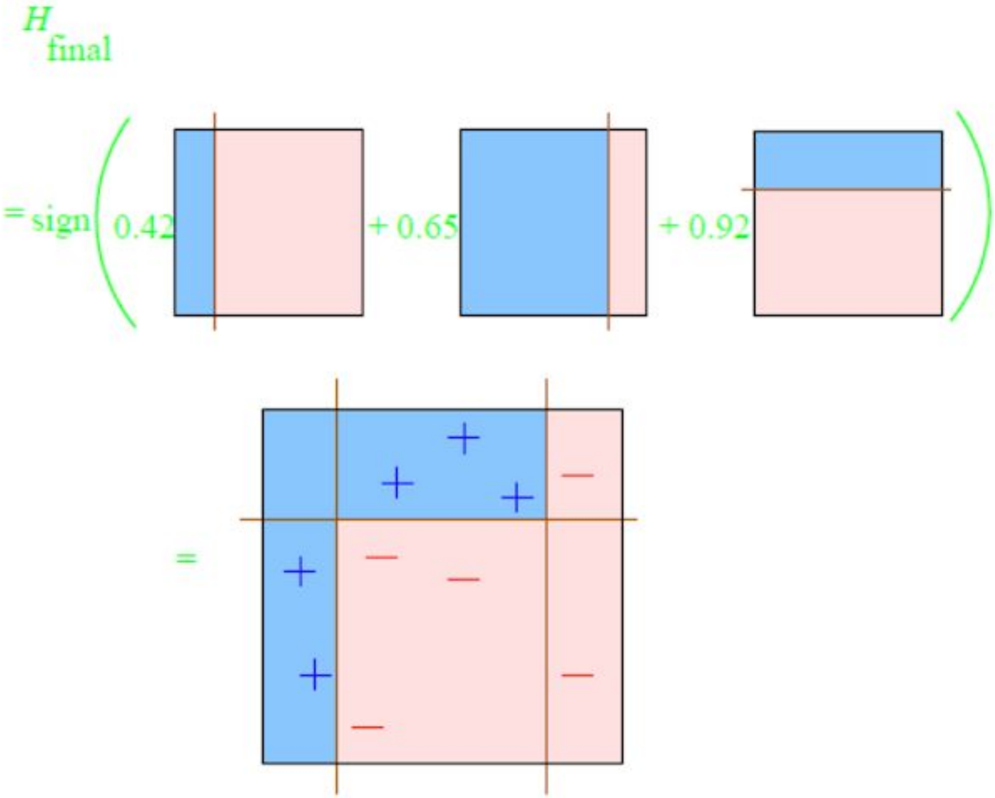
D_{t+1} puts half of weight on examples x_i where h_t is incorrect & half on examples where h_t is correct

Final hyp: $H_{final}(x) = \text{sign}(\sum_t \alpha_t h_t(x))$

Example


D_1
D_2
D_3
h_1
h_2
h_3
ε_1=0.30
α_1=0.42
ε_2=0.21
α_2=0.65
ε_3=0.14
α_3=0.92

Example



Adaboost

- Very general: a meta-procedure, it can use any weak learning algorithm!!! (e.g., Naïve Bayes, decision stumps)
- Very fast (single pass through data each round) & simple to code, no parameters to tune.
- Shift in mindset: goal is now just to find classifiers a bit better than random guessing.
- Grounded in rich theory.
- Relevant for big data age: quickly focuses on "core difficulties", well-suited to distributed settings, where data must be communicated efficiently [Balcan-Blum-Fine-Mansour COLT'12].

Theoretical Guarantees

Theorem $err_S(H_{final}) \leq \exp\left[-2\sum_t \gamma_t^2\right]$ where $\epsilon_t = 1/2 - \gamma_t$

How about generalization guarantees?

Original analysis [Freund&Schapire'97]

- Let H be the set of rules that the weak learner can use
- Let G be the set of weighted majority rules over T elements of H (i.e., the things that AdaBoost might output)

Theorem [Freund&Schapire'97]

$$\forall\, g \in G, err(g) \leq err_S(g) + \tilde{O}\left(\sqrt{\frac{Td}{m}}\right)$$

T= # of rounds

d= VC dimension of H

Questions?

- Questions on Adaboost and boosting?
- Questions on HW4?