6.034

Boosting

Adaboost

Randall Davis

Learning

- Nearest neighbors, near misses, neural nets,…
  - Single approximations to the problem
- Boosting
  - Multiple methods
  - … accumulated incrementally
  - … moving us from weak classifiers to strength in numbers
  - Adaboost
  - Empirical performance

Getting Started

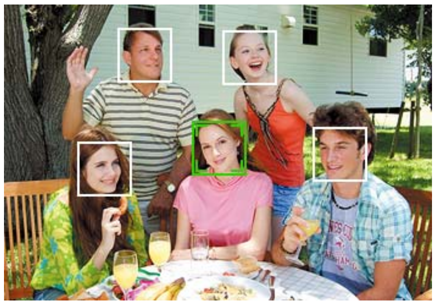
- Binary classification problem?
- Weak classifier?
  - $\varepsilon < 0.5$
- Why would multiple not-so good elements add up to something better?

An Intuition

- Informal soccer (aka football) game with
  - people you don't know
  - who are uniformly not very good in general
- How do you select people for your team?
- How do you select a team?

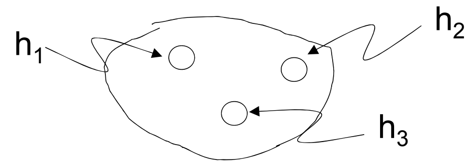
More Realistic Problem

- Face detection



Refining the Intuition

- A set of weak binary classifiers: $h_1, h_2, h_3, \ldots$
- Majority wins: $H(x) = \text{sign}(h_1(x) + h_2(x) + h_3(x))$



- $H(x) = \text{sign}(\alpha_1 h_1(x) + \alpha_2 h_2(x) + \alpha_3 h_3(x))$

Adaboost

- The ultimate excuse for a committee – how a bunch of mediocre people can add up to smart
- Multiple rounds of classifier selection, with training instances re-weighted at each round *to emphasize the errors*
- Can be used to learn a good classifier
- Final classification based on weighted vote of multiple *weak classifiers*
  - weak: < 50% error over any distribution
  - (ie if you're better than a coin flip, you can be on the committee)

7

Adaboost, Formally

- given training set $(x_1, y_1), \ldots, (x_m, y_m)$
- $y_i \in \{-1, +1\}$ correct label of instance $x_i \in X$
- for $t = 1, \ldots, T$:
  - construct distribution D_t on $\{1, \ldots, m\}$
  - find weak hypothesis ("rule of thumb")

    $h_t : X \to \{-1, +1\}$

    with small error ϵ_t on D_t:

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

Adaboost, Formally

- constructing D_t:
  - $D_1(i) = 1/m$
  - given D_t and h_t:

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

$$= \frac{D_t(i)}{Z_t} \cdot \exp(-\alpha_t\, y_i\, h_t(x_i))$$

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

Vorpal Sword

Adaboost, Formally

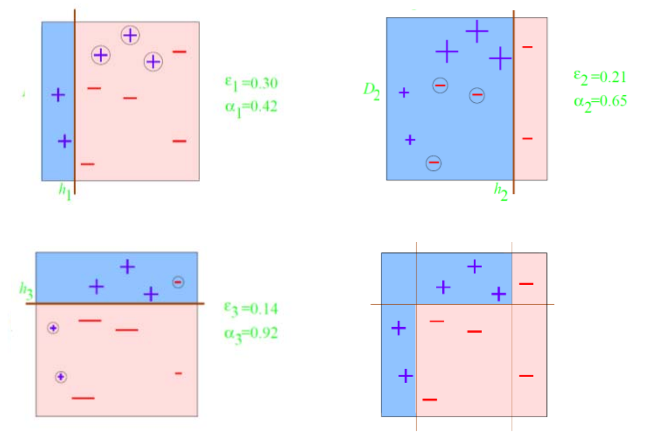
- constructing D_t:
  - $D_1(i) = 1/m$
  - given D_t and h_t:

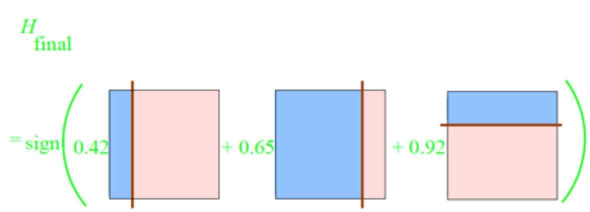
$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$

$$= \frac{D_t(i)}{Z_t} \cdot \exp(-\alpha_t \, y_i \, h_t(x_i))$$

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

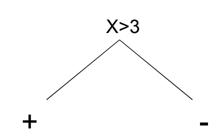
$$\varepsilon^t = \sum_{wrong} D_t \qquad Z_t = 2\sqrt{\epsilon^t(1-\epsilon^t)}$$



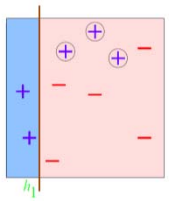


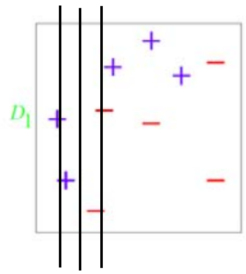
Whence the h_i's?

- Most anywhere
- One easy answer: stumps
  - Single-level decision trees



Stumps





Generality of Adaboost

- What are the h_i?

Taming The Math

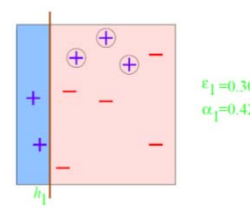
- Updating weights
  - Turns out that for correct answers: $\sum D_i^t = 1/2$

    Scale wts on correct answers *down* to 0.5

  - For wrong answers: $\sum D_i^t = 1/2$

    Scale wts on correct answers *up* to 0.5

Taming The Math



Original weights: 0.1
Correct ans: 7, sums to 0.7
Multiply by 5/7 to scale sum to .5; get new weights of 5/7 * 0.1 = 0.071
Incorrect ans: 3, sums to 0.3,
Multiply by 5/3 to scale sum to .5; get new weights of 5/3 * 0.1 = 0.167

Ada-Boost Summary

- Starting with a Training Set (initial weights 1/n)
  - Weak learning algorithm returns a classifier
  - Reweight the examples
    - Weight on correct examples is decreased
    - Weight on errors is increased
- Final classifier is a weighted majority of Weak Classifiers
  - Classifiers with low error get larger weight

What's So Good About Adaboost

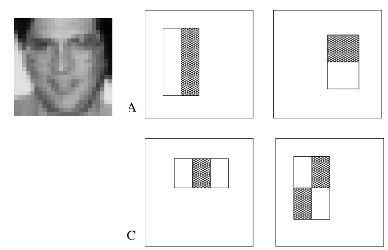
- Improves classification accuracy
- Can be used with many different classifiers
- Commonly used in many areas
- Simple to implement
- Not prone to overfitting
- Speed

An Early Application

- Viola/Jones Face Detection

Image Features

"Rectangle filters"



Differences between sums of pixels in adjacent rectangles

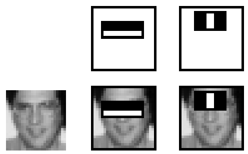
$$h_t(x) = \begin{cases} +1 & \text{if } f_t(x) > q_t \\ -1 & \text{otherwise} \end{cases}$$

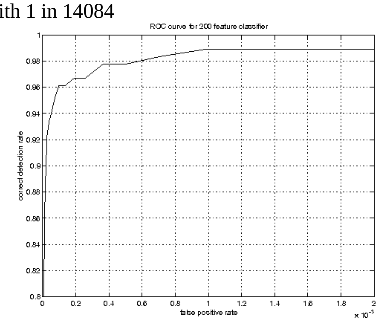
Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

Example Classifier for Face Detection

A classifier with 200 rectangle features was learned using AdaBoost

95% correct detection on test set with 1 in 14084 false positives.





ROC curve for 200 feature classifier

Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001