Ensemble Methods: Boosting

Instructor: Jessica Wu -- Harvey Mudd College

The instructor gratefully acknowledges Eric Eaton (UPenn), Jenna Wiens (UMich), Tommi Jaakola (MIT), David Kauchak (Pomona), David Sontag (NYU), Piyush Rai (Utah), and the many others who made their course materials freely available online.

Boosting

Learning Goals

- Describe boosting
  - How does boosting improve performance?
- Describe the AdaBoost algorithm
- Describe the loss function for AdaBoost

More Tutorials
Robert Schapire (one of the original authors of AdaBoost): http://rob.schapire.net/papers/explaining-adaboost.pdf
Gentler overview: http://mccormickml.com/2013/12/13/adaboost-tutorial/

Ensemble Learning

Bagging reduces variance by averaging. Bias did not change.

Can we reduce bias and variance? Yes, boosting!

Boosting: Combine simple "weak" base learners into a more complex "strong" ensemble.

Insight

- Easy to find "rules of thumb" that are "often" correct
- Hard to find single highly accurate prediction rule

Approach

- Devise program for deriving rough rules of thumb
- Apply procedure to subset of examples, obtain rule of thumb
- Repeat previous step

Technical Details

Assume we are given a "weak" learning algorithm that can consistently find classifiers ("rules of thumb") at least slightly better than random (accuracy > 50% in two-class setting).

Then given sufficient training data, a boosting algorithm can provably construct single classifier with very high accuracy.

Strong and Weak Learnability

Boosting's roots are in "PAC" (probably approximately correct) learning model

"strong" learner

- Given polynomially many training examples (and polynomial time)
  target error rate ε
  failure probability p
- Produce classifier with arbitrarily small generalization error (error rate < ε)
  with high probability ($1-p$)

"weak" learner

- Given polynomially many training examples (and polynomial time)
  failure probability p
- Produce classifier that is slightly better than random guessing (error rate < 0.5)
  with high probability ($1-p$)

Weak learners are much easier to create!
Combine weak learners into strong learner!

Based on slide by Rob Schapire

Key Details

How do we choose examples each round?

- Concentrate on "hardest examples"
  (those most often misclassified by previous rules)

How do we combine rules of thumb into single prediction rule?

- Take (weighted) majority vote of rules of thumb

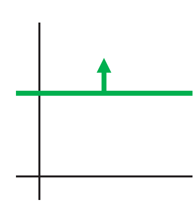
How do we choose weak classifiers?

- Use decision stumps $h(\boldsymbol{x}, \boldsymbol{\theta}) = \mathrm{sgn}(\theta_1^s x_k + \theta_0^s)$
  where $\boldsymbol{\theta} = \left([\theta_0^s, \theta_1^s], k\right)$

encodes location

encodes direction of stump (positive, negative)

encodes coordinate k that stump depends on



Based on slide by Rob Schapire

Boosting Overview

Training

- Start with equal example weights
- For some number of iterations
  - Learn weak classifiers and save
  - Change example weights

Prediction

- Get prediction from all weak classifiers
- Make weighted vote based on how well weak classifier did when it was trained

Based on slide by David Kauchak

Adaboost (adaptive boosting) Algorithm

Set $\widetilde{W}_0^{(i)} = \frac{1}{n}$ for $i = 1, \ldots, n$

For stage $t = 1, \ldots, m$, do

Fit classifier $h(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t)$ to weighted training set (weights $\widetilde{W}_{t-1}$)

Compute weighted classification error

$$\hat{\varepsilon}_t = \sum_{i=1}^{n} \widetilde{W}_{t-1}^{(i)} \mathbb{I}\left[\left[y^{(i)} \neq h\left(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{\theta}}_t\right)\right]\right]$$

Compute "score" $\hat{\alpha}_t = \frac{1}{2} \ln \frac{1-\hat{\varepsilon}_t}{\hat{\varepsilon}_t}$ (ln = natural log; new component is assigned vote based on error)

Update weights on all training examples

$$\widetilde{W}_t^{(i)} = c_t \widetilde{W}_{t-1}^{(i)} \exp\left(-y^{(i)} \hat{\alpha}_t h\left(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{\theta}}_t\right)\right)$$

(where c_t is normalization constant to ensure weights $\widetilde{W}_t^{(i)}$ sum to 1)

Return $h_m(\boldsymbol{x}) = \sum_{t=1}^{m} \hat{\alpha}_t h\left(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t\right)$

Understanding Adaboost

Set $\widetilde{W}_0^{(i)} = \frac{1}{n}$ for $i = 1, \ldots, n$

For stage $t = 1$

$\widetilde{W}_t^{(i)}$ is a vector of weights over the examples at stage t. All points start with equal weight.

Fit classifie

Compute weighted classification error

$$\hat{\varepsilon}_t = \sum_{i=1}^{n} \widetilde{W}_{t-1}^{(i)} \mathbb{I}\left[\left[y^{(i)} \neq h\left(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{\theta}}_t\right)\right]\right]$$

Compute "score" $\hat{\alpha}_t = \frac{1}{2} \ln \frac{1-\hat{\varepsilon}_t}{\hat{\varepsilon}_t}$ (ln = natural log; new component is assigned vote based on error)

Update weights on all training examples

$$\widetilde{W}_t^{(i)} = c_t \widetilde{W}_{t-1}^{(i)} \exp\left(-y^{(i)} \hat{\alpha}_t h\left(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{\theta}}_t\right)\right)$$

(where c_t is normalization constant to ensure weights $\widetilde{W}_t^{(i)}$ sum to 1)

Return $h_m(\boldsymbol{x}) = \sum_{t=1}^{m} \hat{\alpha}_t h\left(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t\right)$

Understanding Adaboost

Set $\widetilde{W}_0^{(i)} = \frac{1}{n}$ for $i = 1, \ldots, n$

For stage $t = 1, \ldots, m$, do

Fit classifier $h(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t)$ to weighted training set (weights $\widetilde{W}_{t-1}$)

Compute weighted

We need a classifier that can be trained with weighted examples.

The training algorithm must be fast (since a new classifier is trained at every stage).

Compute "score"

Update weights on

$$\widetilde{W}_t^{(i)} = c_t \widetilde{W}_{t-1}^{(i)} \exp\left(-y^{(i)} \hat{\alpha}_t h\left(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{\theta}}_t\right)\right)$$

(where c_t is normalization constant to ensure weights $\widetilde{W}_t^{(i)}$ sum to 1)

Return $h_m(\boldsymbol{x}) = \sum_{t=1}^{m} \hat{\alpha}_t h\left(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t\right)$

Understanding Adaboost

Set $\widetilde{W}_0^{(i)} = \frac{1}{n}$ for $i = 1, \ldots, n$

For stage $t = 1, \ldots, m$, do

Fit classifier $h(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t)$ to weighted training set (weights $\widetilde{W}_{t-1}$)

Compute weighted classification error

$$\hat{\varepsilon}_t = \sum_{i=1}^{n} \widetilde{W}_{t-1}^{(i)} \mathbb{I}\left[\left[\underbrace{y^{(i)} \neq h\left(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{\theta}}_t\right)}_{\text{misclassified example}}\right]\right]$$

Error is a weighted sum of all misclassified examples.
Error is between 0 (all examples correctly classified) and 1 (all examples incorrectly classified).

(where c_t is normalization constant to ensure weights $\widetilde{W}_t^{(i)}$ sum to 1)

Return $h_m(\boldsymbol{x}) = \sum_{t=1}^{m} \hat{\alpha}_t h\left(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t\right)$

Understanding Adaboost

Set $\widetilde{W}_0^{(i)} = \frac{1}{n}$ for $i = 1, \ldots, n$

For stage $t = 1, \ldots, m$, do

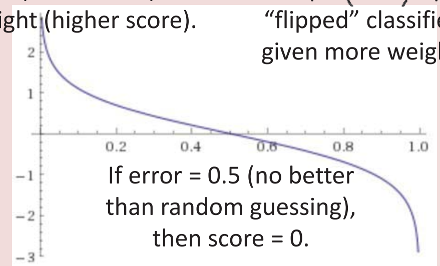
Fit classifier $h(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t)$ to weighted training set (weights $\widetilde{W}_{t-1}$)

$\hat{\alpha}_t$ measures the importance of $h\left(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{\theta}}_t\right)$.
What does it look like (as a function of $\hat{\varepsilon}_t$)?

Compute "score" $\hat{\alpha}_t = \frac{1}{2} \ln \frac{1-\hat{\varepsilon}_t}{\hat{\varepsilon}_t}$ (ln = natural log; new component is assigned vote based on error)

If error < 0.5, then score > 0.
Better classifiers (lower error) are given more weight (higher score).

If error > 0.5, then score < 0.
Flip $h\left(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{\theta}}_t\right)$'s predictions. Better "flipped" classifiers (higher error) are given more weight (higher abs score).



If error = 0.5 (no better than random guessing), then score = 0.

Understanding Adaboost

Set $\widetilde{W}_0^{(i)} = \frac{1}{n}$ for $i = 1, \ldots, n$

For stage $t = 1, \ldots, m$, do

This is equivalent to

$$\widetilde{W}_t^{(i)} = c_t \widetilde{W}_{t-1}^{(i)} \times \begin{cases} e^{-\hat{\alpha}_t} & \text{if } y^{(i)} = h\left(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{\theta}}_t\right) \\ e^{\hat{\alpha}_t} & \text{if } y^{(i)} \neq h\left(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{\theta}}_t\right) \end{cases}$$

	if $\hat{\alpha}_t > 0$:	if $\hat{\alpha}_t < 0$:
$y^{(i)} = h$	downweight correctly classified examples	upweight correctly classified examples
$y^{(i)} \neq h$	upweight incorrectly classified examples	downweight incorrectly classified examples

Update weights on all training examples

$$\widetilde{W}_t^{(i)} = c_t \widetilde{W}_{t-1}^{(i)} \exp\left(-y^{(i)} \hat{\alpha}_t h\left(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{\theta}}_t\right)\right)$$

(where c_t is normalization constant to ensure weights $\widetilde{W}_t^{(i)}$ sum to 1)

Return $h_m(\boldsymbol{x}) = \sum_{t=1}^{m} \hat{\alpha}_t h\left(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t\right)$

Understanding Adaboost

Set $\widetilde{W}_0^{(i)} = \frac{1}{n}$ for $i = 1, \ldots, n$

For stage $t = 1, \ldots, m$, do

Fit classifier $h(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t)$ to weighted training set (weights $\widetilde{W}_{t-1}$)

Compute weighted classification error

$$\hat{\varepsilon}_t = \sum_{i=1}^{n} \widetilde{W}_{t-1}^{(i)} \mathbb{I}\left[\left[y^{(i)} \neq h\left(\boldsymbol{x}^{(i)}; \hat{\boldsymbol{\theta}}_t\right)\right]\right]$$

Compute "score" $\hat{\alpha}_t = \frac{1}{2} \ln \frac{1-\hat{\varepsilon}_t}{\hat{\varepsilon}_t}$ (ln = natural log; new component is assigned vote based on error)

Predict using weighted vote of component classifiers. Remember, better classifiers (or flipped classifiers) are given more weight.

Return $h_m(\boldsymbol{x}) = \sum_{t=1}^{m} \hat{\alpha}_t h\left(\boldsymbol{x}; \hat{\boldsymbol{\theta}}_t\right)$

Dynamic Behavior of Adaboost

If example is repeatedly misclassified

- Each time, increase its weight
- Eventually, it will be emphasized enough to generate ensemble hypothesis that correctly predicts it

Successive member hypotheses focus on hardest parts of instance space

Based on slide by Eric Eaton

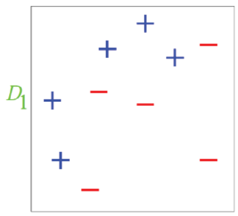
(This slide intentionally left blank.)

Adaboost Example

Consider binary classification with 10 training examples

Determine a boosted combination of decision stumps that correctly classifies all points

Round 0 (initial)



weight distribution is uniform

$$W_0^{(i)} = 1$$

$$\widetilde{W}_0^{(i)} = \frac{1}{10}$$

(This slide intentionally left blank.)

Adaboost Math

Adaboost minimize exponential loss.

$$L(y, \hat{y}) = e^{-y\hat{y}}$$

(Proof? Office Hours)

Other boosting variants

- squared loss $\Rightarrow$ L2-boosting
- absolute error / loss $\Rightarrow$ gradient-boosting
- log loss $\Rightarrow$ logit-boosting

Adaboost in Practice

Pros

- Fast and simple to program
- No parameters to tune (except m)
- No assumptions on weak learner
- Versatile (has been extended to multiclass learning problems)
- Provably effective, provided can consistently find rough rules of thumb

$\Rightarrow$ Shift in mind set

goal now is merely to find classifier barely better than random guessing

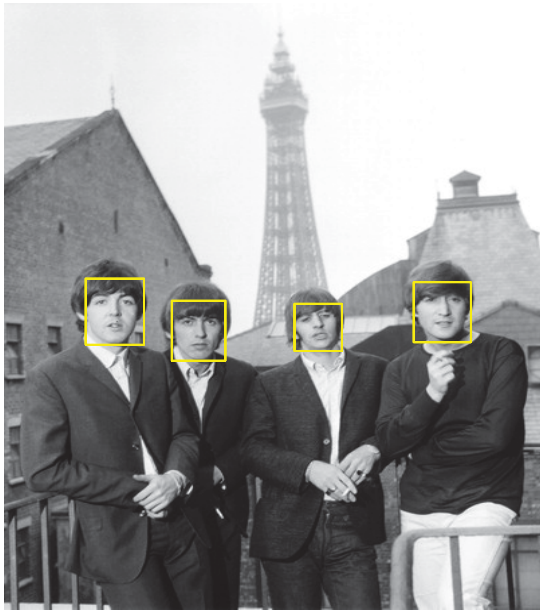
Cons

- Performance depends on weak learner
- Can fail if
  - Weak classifiers too complex $\rightarrow$ overfitting
  - Weak classifiers too weak: insufficient data $\rightarrow$ underfitting; low margins $\rightarrow$ overfitting
- Empirically susceptible to uniform noise

Based on slide by Eric Eaton

Adaboost Application Example

Face detection



Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola
viola@merl.com
Mitsubishi Electric Research Labs
201 Broadway, 8th FL
Cambridge, MA 02139

Michael Jones
mjones@crl.dec.com
Compaq CRL
One Cambridge Center
Cambridge, MA 02142

Rapid object **detection** using a boosted cascade of simple features
P **Viola**, M **Jones** - ... Vision and Pattern Recognition, 2001. CVPR ..., 2001 - ieeexplore.ieee.org
... overlap. Each partition yields a single final **detection**. The ... set. Experiments on a Real-World Test Set We tested our system on the MIT+CMU frontal **face** test set [II]. This set consists of 130 images with 507 labeled frontal **faces**. A ...
Cited by 8422 Related articles All 129 versions Cite Save More

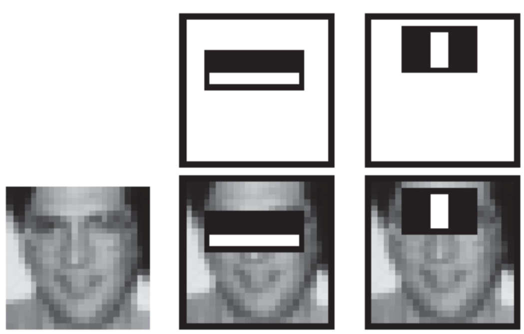
To give you some context of importance…

The anatomy of a large-scale hypertextual Web search engine
S **Brin**, L Page - Computer networks and ISDN systems, 1998 - Elsevier
... This is largely because they all have high **PageRank**. ... However, once the system was running smoothly, S. **Brin**, L. PagelComputer Networks and ISDN Systems 30 ... Google employs a number of techniques to improve search quality including **page rank**, anchor text, and proximity ...
Cited by 11070 Related articles All 349 versions Cite Save

Based on slide by David Kauchak

"Weak" Learners

Detect light / dark rectangles in image



$$h(\boldsymbol{x}) = \alpha_1 h_1(\boldsymbol{x}) + \alpha_2 h_2(\boldsymbol{x}) + \ldots$$
$$h_i(\boldsymbol{x}) = 1 \text{ if } g_i(\boldsymbol{x}) > \theta_i \text{ (threshold)}$$
$$-1 \text{ otherwise}$$
$$g(\boldsymbol{x}) = \text{sum(white\_area)} - \text{sum(black\_area)}$$

Based on slide by David Kauchak

Bagging vs Boosting

Bagging

- Generate random sets from training data
- Combine outputs of multiple classifiers to produce single output
- Decrease variance, bias unaffected

Boosting

- Combine simple “weak” base classifiers into more complex “strong” ensemble
- Decrease bias and variance

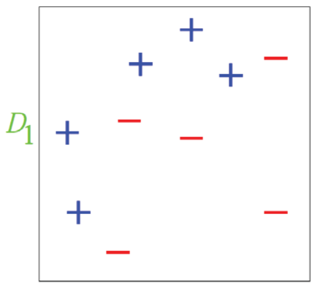
(This slide intentionally left blank.)

Adaboost Example

Consider binary classification with 10 training examples

Determine a boosted combination of decision stumps
that correctly classifies all points

Round 0 (initial)

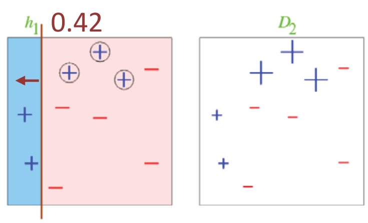


weight distribution is uniform

$$W_0^{(i)} = 1$$
$$\widetilde{W}_0^{(i)} = \frac{1}{10}$$

Adaboost Example

Round 1



$$\hat{\varepsilon}_1 = \frac{3}{10}$$
$$\hat{\alpha}_1 = \frac{1}{2} \ln \frac{1 - \frac{3}{10}}{\frac{3}{10}} = \ln \sqrt{\frac{7}{3}} \approx 0.42$$

each circled point misclassified so upweighted [3 pts]

$$W_1^{(i)} = \frac{1}{10} \exp\left(\ln \sqrt{\frac{7}{3}}\right) \quad = \frac{1}{10}\sqrt{\frac{7}{3}} \approx 0.15 \ \Rightarrow \widetilde{W}_1^{(i)} = \frac{1}{6}$$

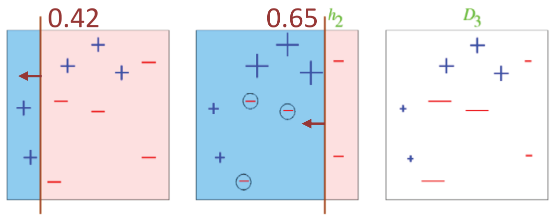
each non-circled point correctly classified point so downweighted [7 pts]

$$W_1^{(i)} = \frac{1}{10} \exp\left(-\ln \sqrt{\frac{7}{3}}\right) = \frac{1}{10}\sqrt{\frac{3}{7}} \approx 0.07 \Rightarrow \widetilde{W}_1^{(i)} = \frac{1}{14}$$

weights then renormalized to 1

Adaboost Example

Round 2



$$\hat{\varepsilon}_2 = 3\left(\frac{1}{14}\right) = \frac{3}{14} \approx 0.21$$

$$\hat{\alpha}_2 = \frac{1}{2}\ln\frac{1-\frac{3}{14}}{\frac{3}{14}} = \ln\sqrt{\frac{11}{3}} \approx 0.65$$

circled – : misclassified so upweighted [3 pts]

$$W_2^{(i)} = \frac{1}{14}\exp\left(\ln\sqrt{\frac{11}{3}}\right) = \frac{1}{14}\sqrt{\frac{11}{3}} \Rightarrow \widetilde{W}_2^{(i)} = \frac{1}{6}$$

large + : correctly classified so downweighted [3 pts]

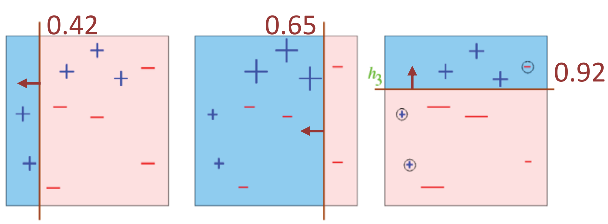
$$W_2^{(i)} = \frac{1}{6}\exp\left(-\ln\sqrt{\frac{11}{3}}\right) = \frac{1}{6}\sqrt{\frac{3}{11}} \Rightarrow \widetilde{W}_2^{(i)} = \frac{7}{66}$$

small + / – : correctly classified so downweighted [4 pts]

$$W_2^{(i)} = \frac{1}{14}\exp\left(-\ln\sqrt{\frac{11}{3}}\right) = \frac{1}{14}\sqrt{\frac{3}{11}} \Rightarrow \widetilde{W}_2^{(i)} = \frac{1}{22}$$

Adaboost Example

Round 3

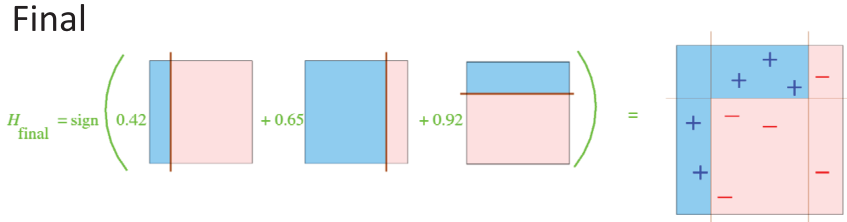


$$\hat{\varepsilon}_3 = 3\left(\frac{1}{22}\right) = \frac{3}{22} \approx 0.14$$

$$\hat{\alpha}_3 = \frac{1}{2}\ln\frac{1-\frac{3}{22}}{\frac{3}{22}} = \ln\sqrt{\frac{19}{3}} \approx 0.92$$

decide to stop after round 3

Final



ensemble consists of 3 classifiers $h_1,\ h_2,\ h_3$

final classifier is weighted linear combination of all classifiers

multiple weak, linear classifiers combined to give strong, nonlinear classifier