CSE 151 Machine Learning

Instructor: Kamalika Chaudhuri

Announcements

Midterm on Monday May 21 (decision trees, kernels, perceptron, and comparison to k-NNs)

Review session on Friday (enter time on Piazza)

Ensemble Learning

How to combine multiple classifiers into a single one

Works well if the classifiers are complementary

This class: two types of ensemble methods:

- Bagging
- Boosting

Boosting

Goal: Determine if an email is spam or not based on text in it

From: Yuncong Chen
Text: 151 homeworks are all graded...

Not Spam

From: Work from home solutions
Text: Earn money without working!

Spam

Sometimes it is:

* Easy to come up with simple rules-of-thumb classifiers,
* Hard to come up with a single high accuracy rule

Boosting

Weak Learner: A simple rule-of-the-thumb classifier that doesn't necessarily work very well

Strong Learner: A good classifier

Boosting: How to combine many weak learners into a strong learner?

Boosting

1. How to get a good rule-of-thumb?

   Depends on application

   e.g, single node decision trees

2. How to choose examples on each round?

   Focus on the **hardest examples** so far **--** namely, examples misclassified most often by previous rules of thumb

3. How to combine the rules-of-thumb to a prediction rule?

   Take a weighted majority of the rules

Some Notation

Let D be a distribution over examples, and h be a classifier
Error of h with respect to D is:

$$err_D(h) = Pr_{(X,Y)\sim D}(h(X) \neq Y)$$

h is called a **weak learner** if $err_D(h) < 0.5$

If you guess completely randomly, then the error is 0.5

Boosting

Given training set $S = \{(x_1, y_1), ..., (x_n, y_n)\}$, y in {-1, 1}

For t = 1, ..., T

Construct distribution D_t on the examples

Find weak learner h_t which has small error $err_{Dt}(h_t)$ wrt D_t

Output final classifier

Initially, $D_1(i) = 1/n$, for all i (uniform)

Given D_t and h_t:

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

where:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - err_{D_t}(h_t)}{err_{D_t}(h_t)} \right)$$

Z_t = normalization constant

Boosting

Given training set $S = \{(x_1, y_1), ..., (x_n, y_n)\}$, y in {-1, 1}

For t = 1, ..., T

Construct distribution D_t on the examples

Find weak learner h_t which has small error $err_{Dt}(h_t)$ wrt D_t

Output final classifier

Initially, $D_1(i) = 1/n$, for all i (uniform)

Given D_t and h_t:

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

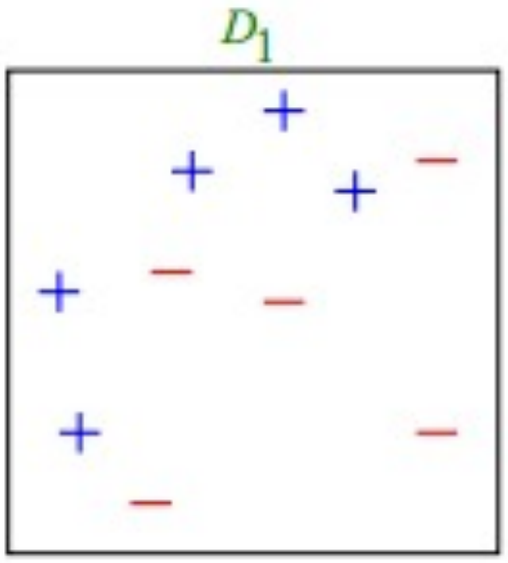
where:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - err_{D_t}(h_t)}{err_{D_t}(h_t)} \right)$$

Z_t = normalization constant

Final classifier: $sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

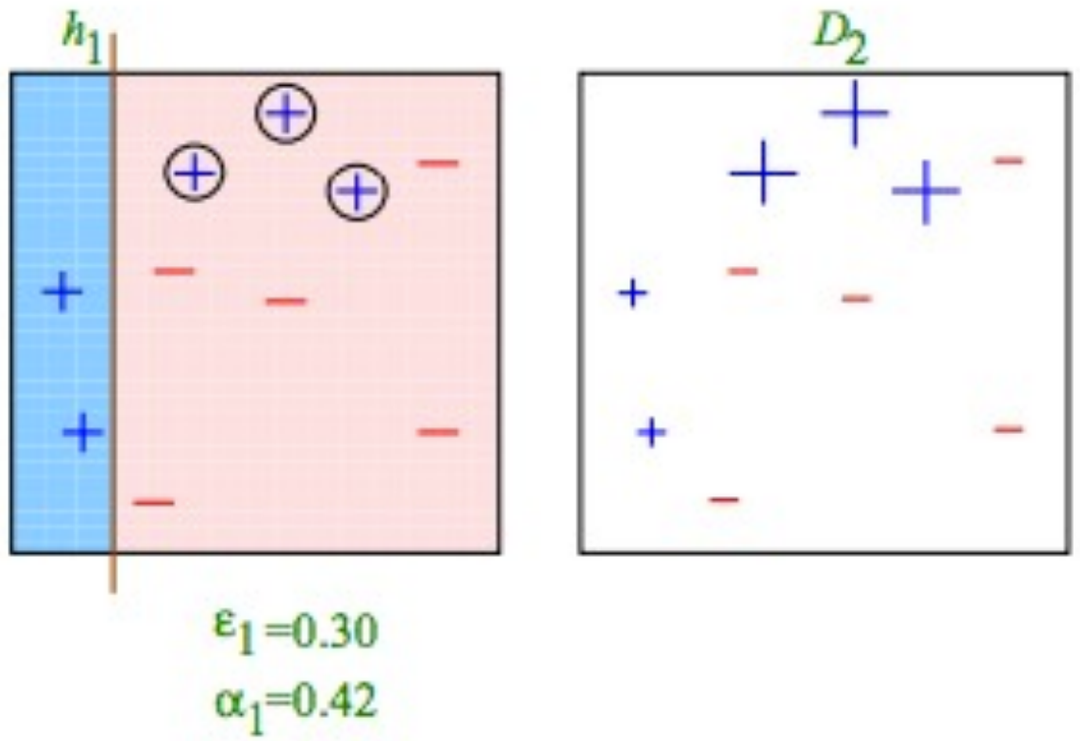
Boosting: Example



Schapire, 2011

weak classifiers: horizontal or vertical half-planes

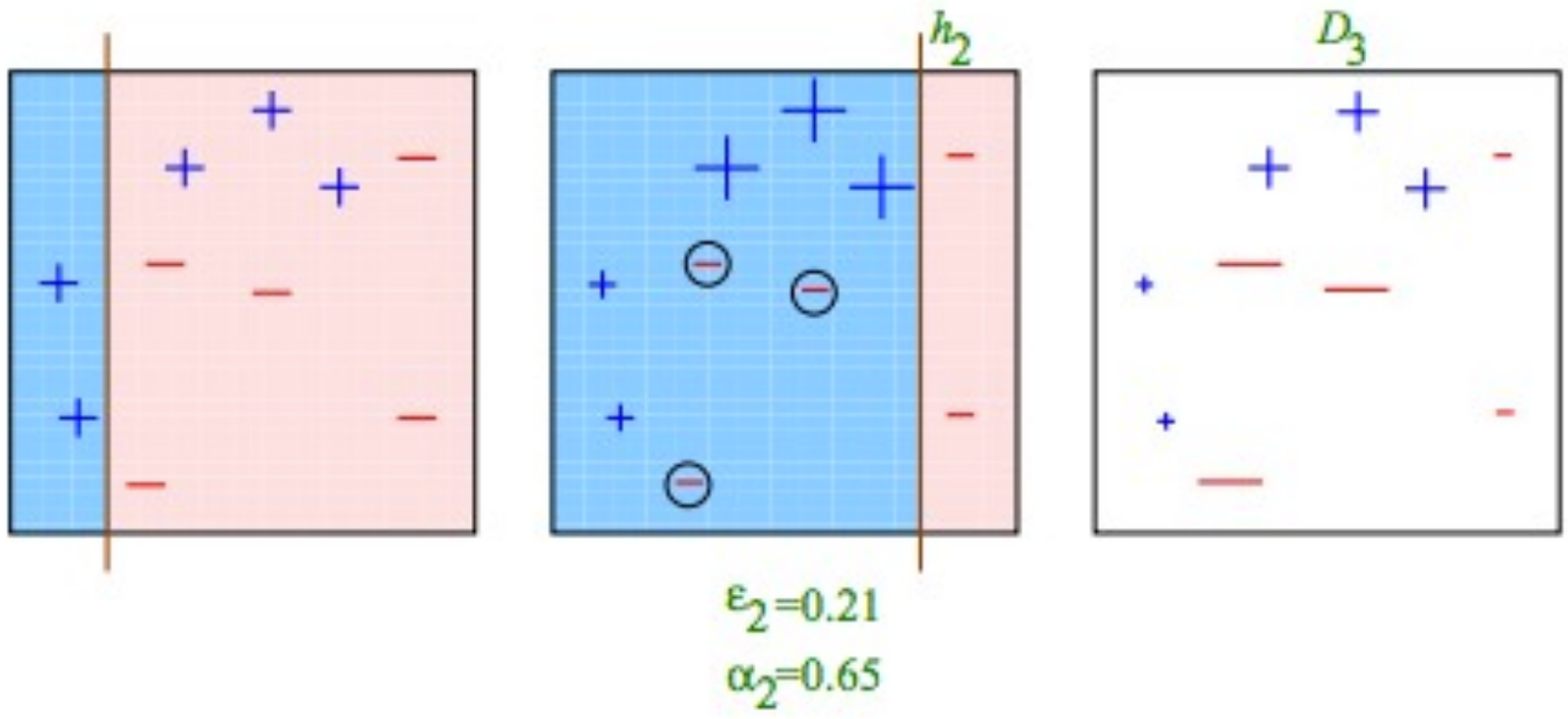
Boosting: Example



Schapire, 2011

weak classifiers: horizontal or vertical half-planes

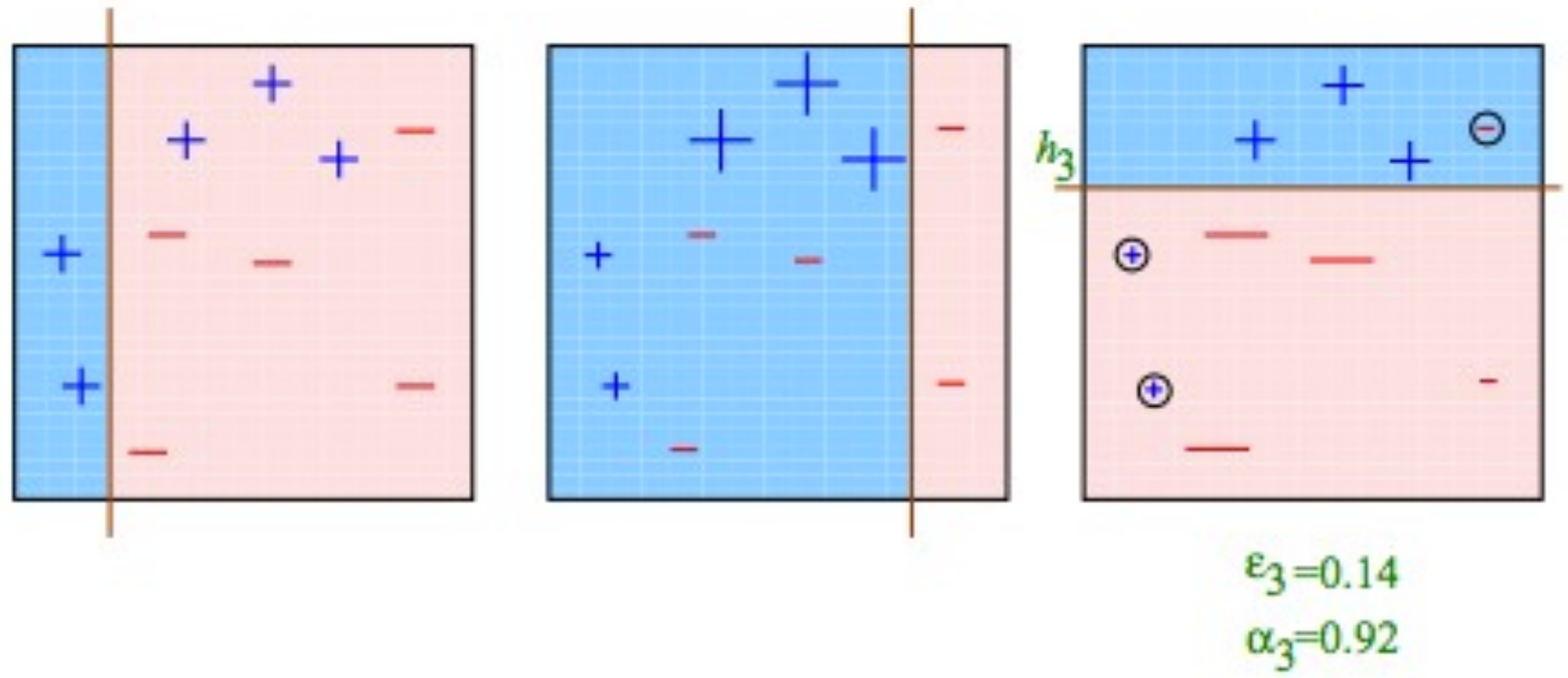
Boosting: Example



Schapire, 2011

weak classifiers: horizontal or vertical half-planes

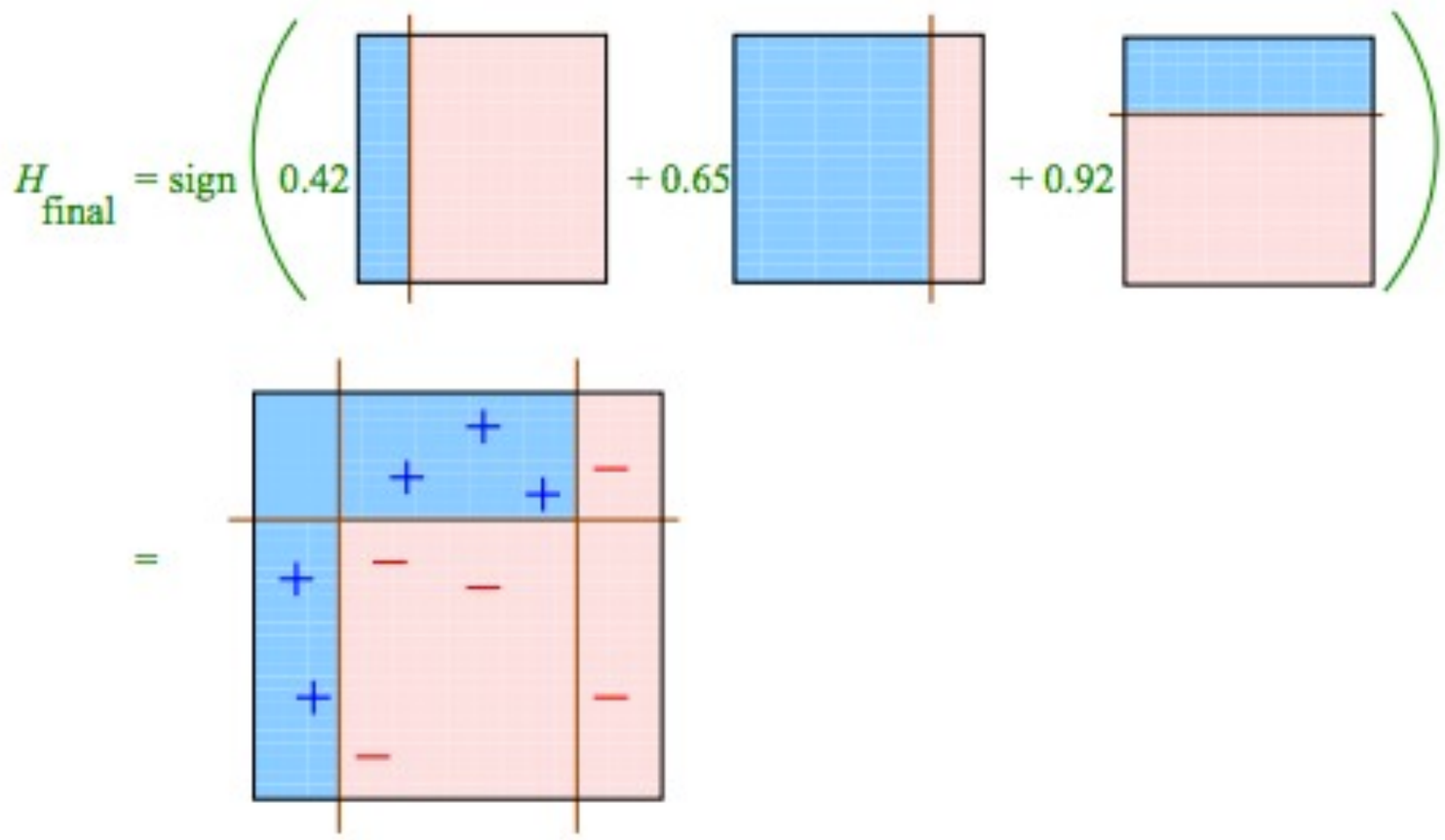
Boosting: Example



Schapire, 2011

weak classifiers: horizontal or vertical half-planes

The Final Classifier



Schapire, 2011

weak classifiers: horizontal or vertical half-planes

How to Find Stopping Time

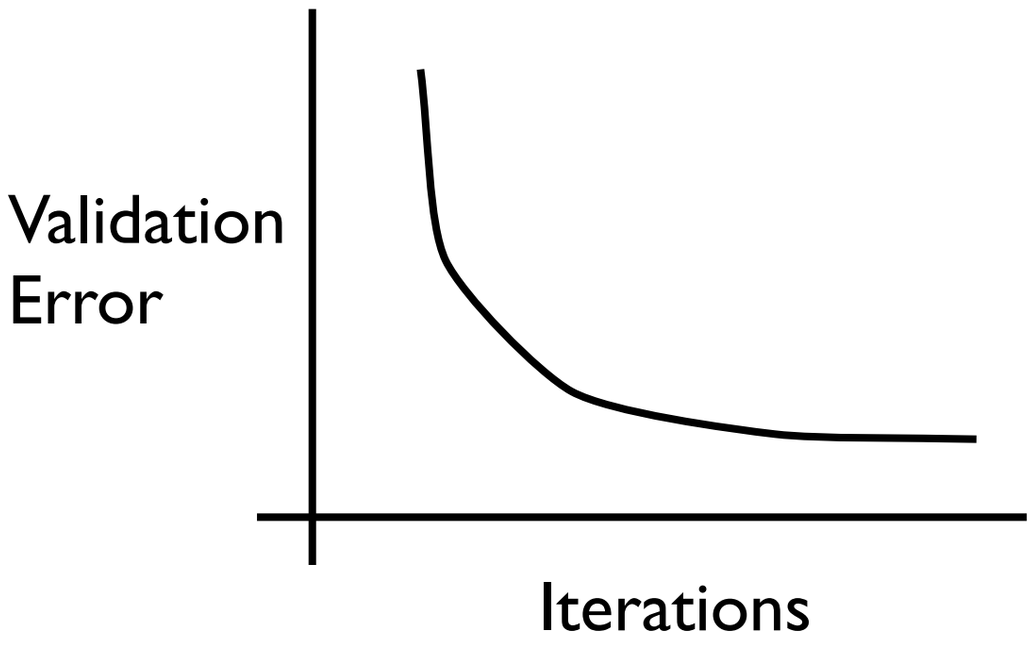
Given training set $S = \{(x_1, y_1), ..., (x_n, y_n)\}$, y in {-1, 1}

For t = 1, ..., T

Construct distribution D_t on the examples

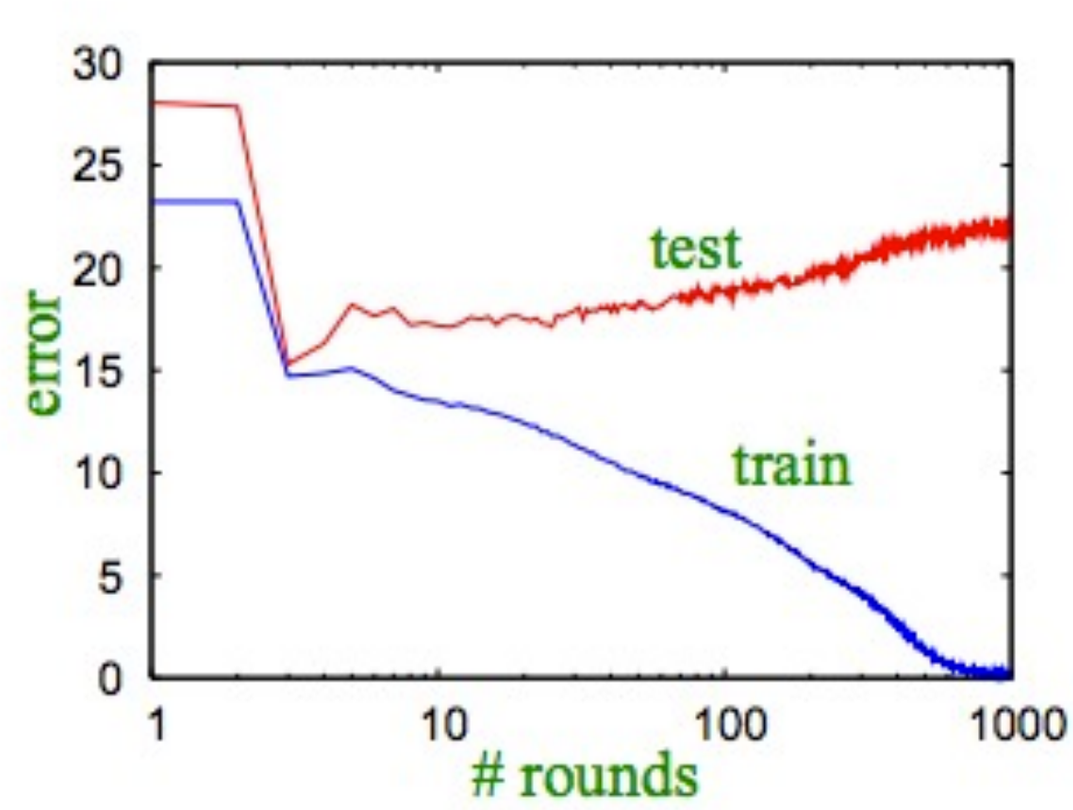
Find weak learner h_t which has small error $err_{Dt}(h_t)$ wrt D_t

Output final classifier



To find stopping time, use a **validation dataset**.
Stop when the error on the validation dataset stops getting better, or when you can't find a good rule of thumb.

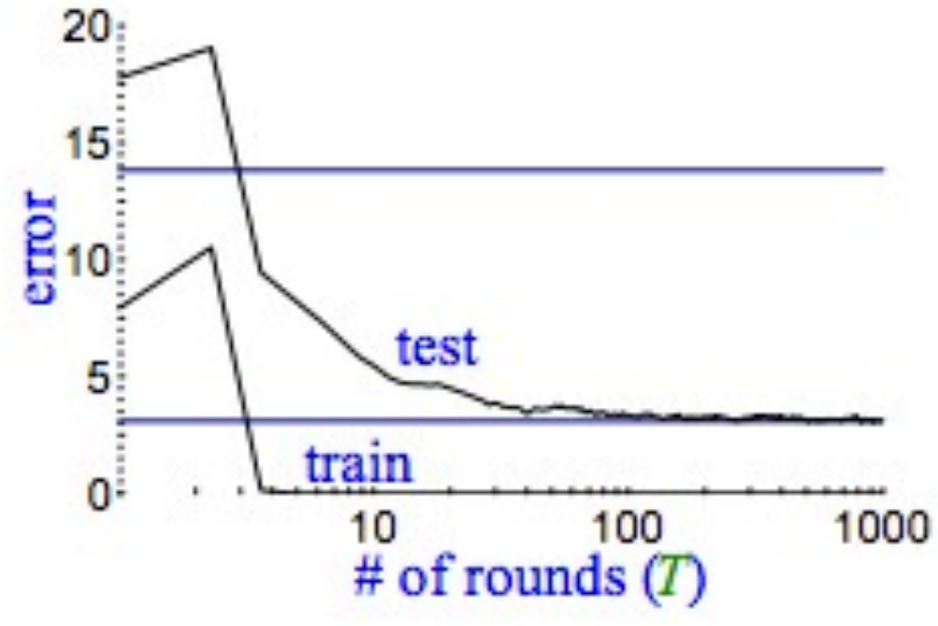
Boosting and Overfitting



Boosting single node decision trees on heart disease dataset

Overfitting can happen with boosting, but often doesn't

Typical Boosting Run

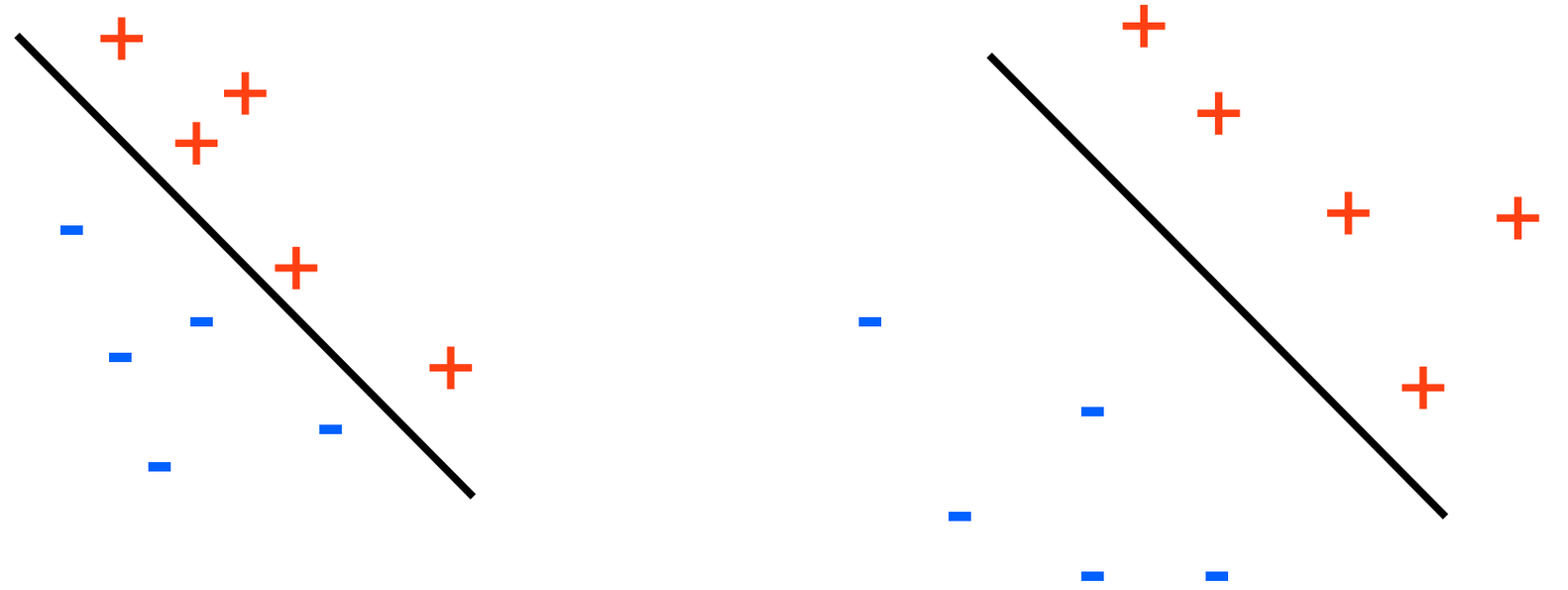


Boosting decision trees
on letter dataset

The reason is that the **margin of classification** often increases with boosting

Margin

Intuitively, margin of classification measures how far the + labels are from the - labels

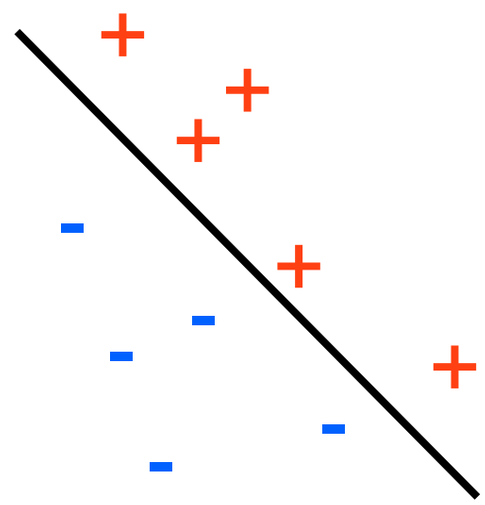


Small Margin **Large Margin**

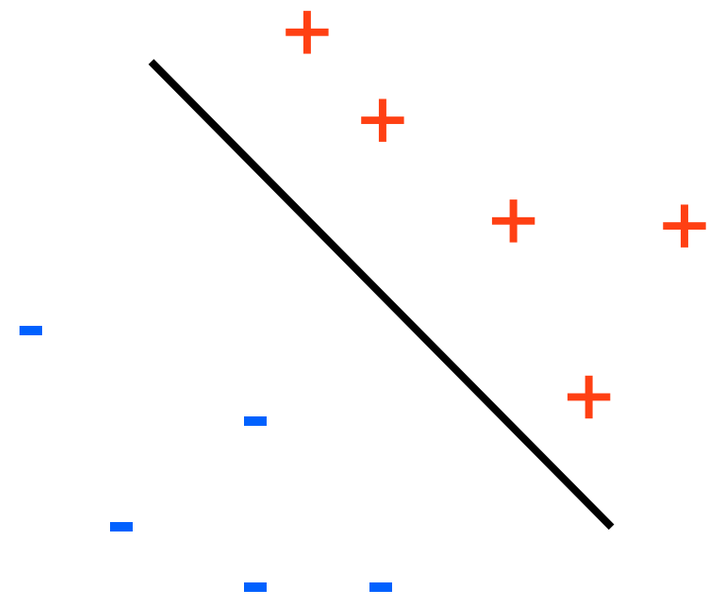
For boosting, margin of example x is: $|\sum_{t=1}^{T} \alpha_t h_t(x)|$

Margin

Intuitively, margin of classification measures how far the + labels are from the - labels



Small Margin



Large Margin

Large margin classifiers generalize more easily than small margin ones

(This is also the reason why kernels work sometimes!)

Advantages and Disadvantages

1. Fast
2. Simple (easy to program)
3. No parameters to tune (other than the stopping time)
4. Provably effective, provided we can find good weak learners

Application: Face Detection (Viola-Jones'00)

Given a rectangular window of pixels, is there a face in it?

Properties:

* Easy to come up with simple rules-of-thumb classifiers,
* Hard to come up with a single high accuracy rule

Viola-Jones Weak Learners

A weak learner $h_{f,t,s}$ is described by:

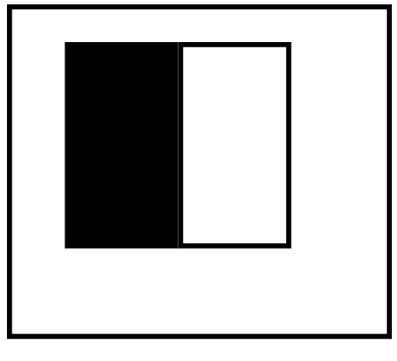
* feature f
* threshold t
* sign s (+1 or -1)

For an example x,

$$h_{f,t,s}(x) = 1, \text{if } sf(x) \geq t$$
$$= -1, \text{otherwise}$$

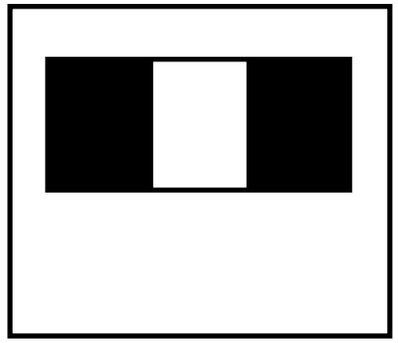
Viola-Jones: 3 Types of Features

1



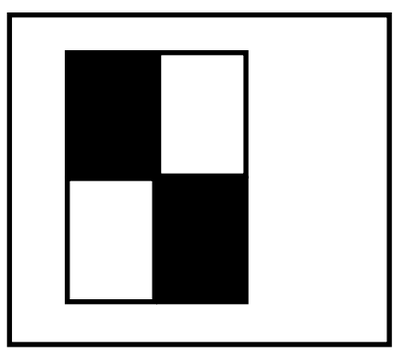
Feature value = sum of pixel colors in black rectangle - sum of pixel colors in white rectangle

2



Feature value = sum of pixel colors in black rectangles - sum of pixel colors in white rectangle

3



Feature value = sum of pixel colors in black rectangles - sum of pixel colors in white rectangles

Viola-Jones Weak Learners

A weak learner $h_{f,t,s}$ is described by:

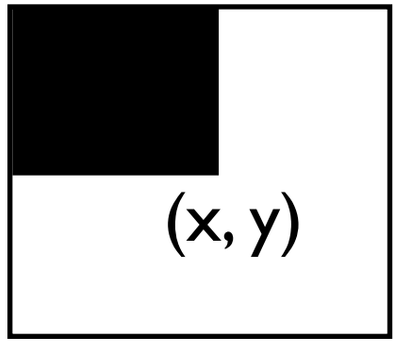
* feature f (3 types of rectangular features)
* threshold t
* sign s (+1 or -1)

For an example x,

$$h_{f,t,s}(x) = 1, \text{if } sf(x) \geq t$$
$$= -1, \text{otherwise}$$

Viola-Jones: Computing the Features

Precompute and store the values s(x,y) for each (x, y):



s(x, y) = sum of pixel colors in the black rectangle

Now each feature can be computed from adding/subtracting a constant number of s(x,y)'s

Viola-Jones: Procedure

Given training set S = $\{(x_1, y_1), ..., (x_n, y_n)\}$, y in {-1, 1}

For t = 1, ..., T

Construct distribution D_t on the examples

Find weak learner h_t which has small error $err_{Dt}(h_t)$ wrt D_t

Output final classifier

Weak learning procedure: Find the feature f, sign s, and threshold t for which the error of $h_{f,t,s}$ on D_t is minimum

Viola and Jones: Procedure

Given training set S = $\{(x_1, y_1), ..., (x_n, y_n)\}$, y in {-1, 1}

For t = 1, ..., T

Construct distribution D_t on the examples

Find weak learner h_t which has small error $err_{Dt}(h_t)$ wrt D_t

Output final classifier

Initially, $D_1(i) = 1/n$, for all i (uniform)

Given D_t and h_t:

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

where:

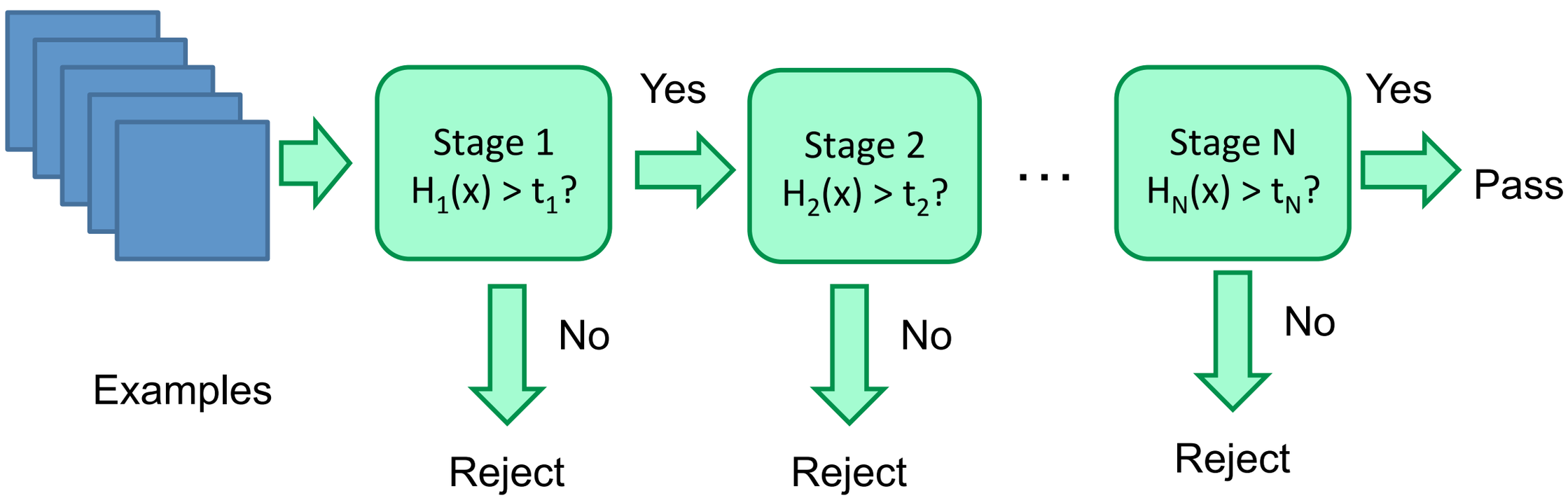
$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - err_{D_t}(h_t)}{err_{D_t}(h_t)} \right)$$

Z_t = normalization constant

Final classifier: $sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

Viola and Jones: Some Results

Cascades for Fast Classification



Choose thresholds for low false negative rates

Fast classifiers earlier in cascade, slower classifiers later

Most examples don't get to the later stages, so system is fast on an average

Boosted Decision Trees

Given training set $S = \{(x_1, y_1), ..., (x_n, y_n)\}$, y in {-1, 1}

For t = 1, ..., T

Construct distribution D_t on the examples

Find weak learner h_t which has small error $err_{Dt}(h_t)$ wrt D_t

Output final classifier

Weak Learners: Single node decision trees

Weak learning process:
Find the single node decision tree that has the lowest error on D_t

Works extremely well in practical applications

Ensemble Learning

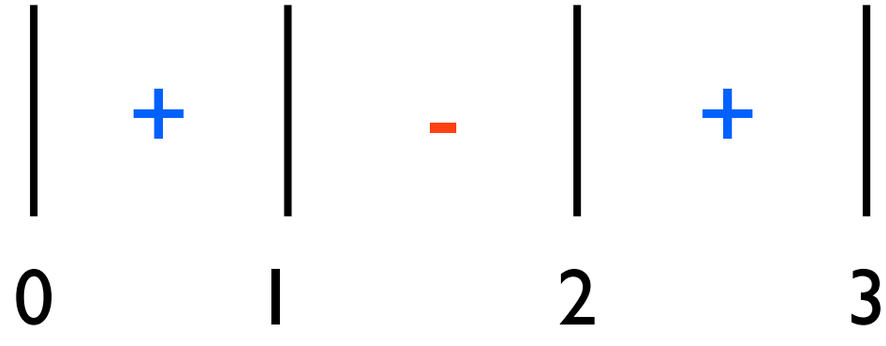
How to combine multiple classifiers into a single one

Works well if the classifiers are complementary

This class: two types of ensemble methods:

- Bagging
- Boosting

In Class Problem



Weak Learners: Thresholds h_{0+}, h_{1+}, h_{2+}, h_{3+},h_{0-}, h_{1-}, h_{2-}, h_{3-}

$h_{i+}(x) = +$, if $x > i$, - otherwise

$h_{i-}(x) = +$, if $x < i$, - otherwise

(1) What is the best classifier in this set? What is its error?

(2) What is the error of sign($h_{0+}(x) + h_{2+}(x) + h_{1-}(x)$)?