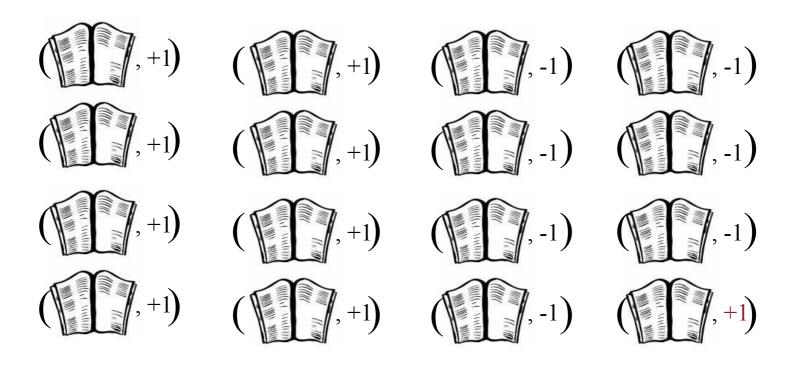
Introduction to Boosting

Cynthia Rudin PACM, Princeton University



Say you have a database of news articles...



where articles are labeled '+1' if the category is "entertainment", and '-1' otherwise.

Your goal is: Given a new article



, find its label.

Examples of Statistical Learning Tasks:

- Optical Character Recognition (OCR) (post office, banks), object recognition in images.
- Bioinformatics (analysis of gene array data for tumor detection, protein classification, etc.)
- Webpage classification (search engines), email filtering, document retrieval
- Semantic classification for speech, automatic .mp3 sorting
- Time-series prediction (regression)

Examples of classification algorithms:

- SVM's (Support Vector Machines large margin classifiers)
- Neural Networks
- Decision Trees / Decision Stumps (CART)
- RBF Networks
- Nearest Neighbors
- Bayes Net

Which is the best?

Depends on amount and type of data, and application!

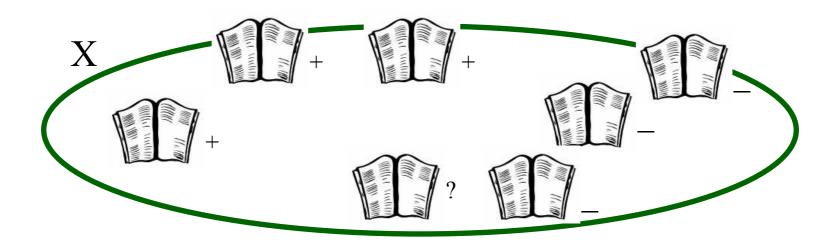
It's a tie between SVM's and **Boosted** Decision Trees/Stumps for general applications.

One can always find a problem where a particular algorithm is the best. Boosted convolutional neural nets are the best for OCR (Yann LeCun et al).

Training Data: $\{(\mathbf{x}_i, y_i)\}_{i=1..m}$ where (\mathbf{x}_i, y_i) is chosen iid from an unknown probability distribution on $X \times \{-1,1\}$.

"space of all possible articles" "labels"

Huge Question: Given a new random example **x**, can we predict its correct label with high probability? That is, can we *generalize* from our training data?



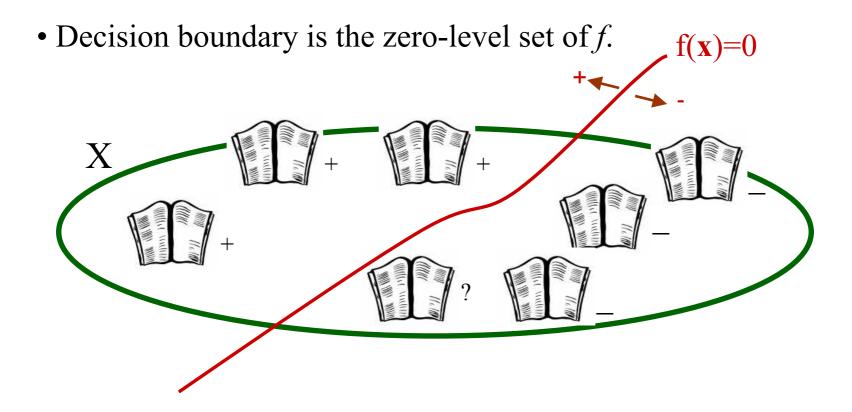
Huge Question: Given a new random example **x**, can we predict its correct label with high probability? That is, can we *generalize* from our training data?

Yes!!! That's what the field of statistical learning is all about.

The goal of statistical learning is to characterize points from an unknown probability distribution when given a representative sample from that distribution.

How do we construct a classifier?

• Divide the space X into two sections, based on the sign of a function $f: X \rightarrow R$.



So Overview of Talk (3)

- The Statistical Learning Problem (done)
- Introduction to Boosting and AdaBoost
- AdaBoost as Coordinate Descent
- The Margin Theory and Generalization

Say we have a "weak" learning algorithm:

- A weak learning algorithm produces weak classifiers.
- (Think of a weak classifier as a "rule of thumb")

Examples of weak classifiers for "entertainment" application:

$$h_1()) = \begin{cases} +1 & \text{if } contains \text{ the term "movie",} \\ -1 & \text{otherwise} \end{cases}$$
 $h_2()) = \begin{cases} +1 & \text{if } contains \text{ the term "actor",} \\ -1 & \text{otherwise} \end{cases}$

$$h_3()$$
 = $\begin{cases} +1 \text{ if } \\ -1 \text{ otherwise} \end{cases}$ contains the term "drama",

Boosting algorithms combine weak classifiers in a meaningful way.

Example:

$$f(\sqrt{1}) = sign[.4 h1 (\sqrt{1}) + .3 h2 (\sqrt{1}) + .3 h3 (\sqrt{1})]$$

ASboidstingartigbrichntainkethæsterput:movie", and the word "drama", but not the word "actor":

- the weak learning algorithm which produces the weak classifiers
- a large training of at abase n[.4-.3+.3] = 1, so we label it +1.

and outputs:

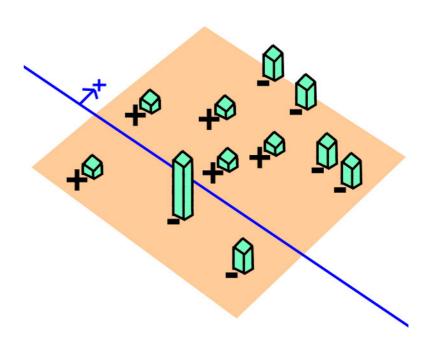
- the coefficients of the weak classifiers to make the combined classifier

Two ways to use a Boosting Algorithm:

- As a way to increase the performance of already 'strong' classifiers.
 - Ex. neural networks, decision trees

- "On their own" with a really basic weak classifier
 - Ex. decision stumps

AdaBoost (Freund and Schapire '95)



- -Start with a uniform distribution ("weights") over training examples.

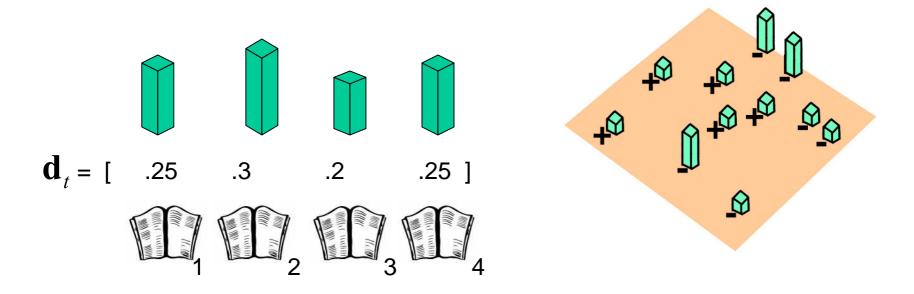
 (The weights tell the regalty parning algorithm which examples are important.) obtained at all iterations.
- -Request a weak classifier from the weak learning algorithm, $h_j: X \to \{-1,1\}$. $f_{\text{final}}(\mathbf{X}) = sign(\lambda_1 h_1(\mathbf{X}) + \ldots + \lambda_n h_n(\mathbf{X}))$
- -Increase the weights on the training examples that were misclassified.

-(Repeat)

AdaBoost

Define three important things:

 $\mathbf{d}_t \in \mathbb{R}^m := \text{distribution ("weights") over examples at time t}$



AdaBoost

Define three important things:

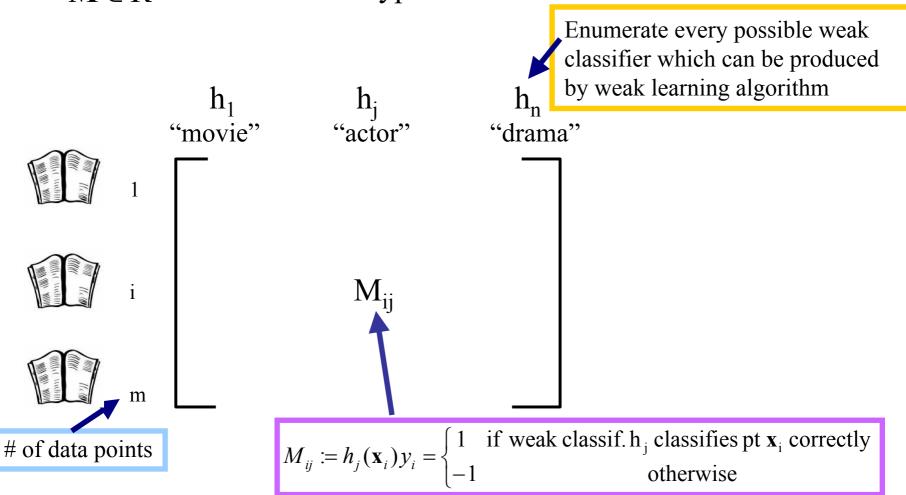
 $\lambda_t \in \mathbb{R}^n := \text{coeffs of weak classifiers for the linear combination}$

$$f_t(\mathbf{x}) = \operatorname{sign}(\lambda_{t,1} h_1(\mathbf{x}) + \dots + \lambda_{t,n} h_n(\mathbf{x}))$$

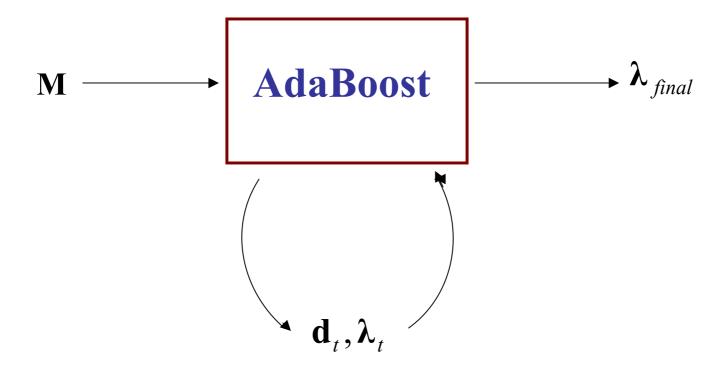
AdaBoost

Define:

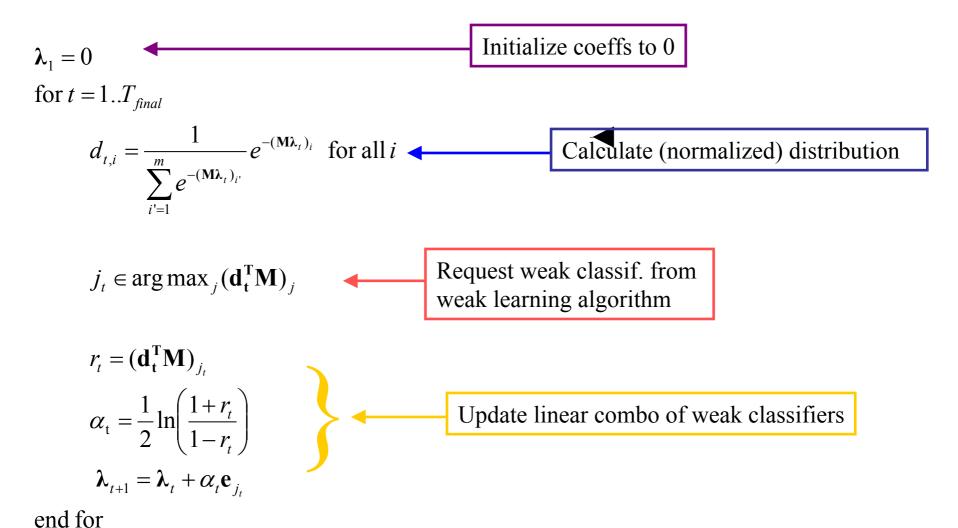
 $\mathbf{M} \in \mathbb{R}^{m \times n}$:=matrix of hypotheses and data



The matrix M has too many columns to actually be enumerated. M acts as the only input to AdaBoost.



AdaBoost (Freund and Schapire 95)



AdaBoost (Freund and Schapire 95)

$$\lambda_1 = 0$$
for $t = 1..T_{final}$

$$d_{t,i} = \frac{1}{\sum_{i'=1}^{m} e^{-(\mathbf{M}\lambda_t)_{i'}}} e^{-(\mathbf{M}\lambda_t)_{i'}} \text{ for all } i$$

$$j_t \in \arg\max_{j} (\mathbf{d_t^T} \mathbf{M})_j$$

 $\lambda_{t+1} = \lambda_t + \alpha_t \mathbf{e}_{j_t}$

"Edge" or "correlation" of weak classifier
$$j_t$$
.
$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 + r_t}{1 - r_t} \right)$$

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 + r_t}{1 - r_t} \right)$$
"Edge" or "correlation" of weak classifier j_t .

end for

AdaBoost as Coordinate Descent

Breiman, Mason et al., Duffy and Helmbold, etc. noticed that AdaBoost is a coordinate descent algorithm.

- Coordinate descent is a minimization algorithm like gradient descent, except that we only move along coordinates.
- We cannot calculate the gradient because of the high dimensionality of the space!
- "coordinates" = weak classifiers "distance to move in that direction" = the update α_t

AdaBoost minimizes the following function via coordinate descent:

$$F(\lambda) := \sum_{i=1}^{m} e^{-(\mathbf{M}\lambda)_i}$$

Choose a direction: $j_t \in \arg\max_{j} (\mathbf{d_t^T M})_j$

Choose a distance to move in that direction:

$$r_{t} = (\mathbf{d}_{t}^{T} \mathbf{M})_{j_{t}}$$

$$\alpha_{t} = \frac{1}{2} \ln \left(\frac{1 + r_{t}}{1 - r_{t}} \right)$$

$$\lambda_{t+1} = \lambda_{t} + \alpha_{t} \mathbf{e}_{j_{t}}$$

The function
$$F(\lambda) := \sum_{i=1}^{m} e^{-(M\lambda)_i}$$
 is convex:

1) If the data is non-separable by the weak classifiers, the minimizer of F occurs when the size of λ is finite.

(This case is ok. AdaBoost converges to something we understand.)

2) If the data is separable, the minimum of F is 0

(This case is confusing!)

The original paper suggested that AdaBoost would probably overfit...

But it didn't in practice!

Why not?

The margin theory!

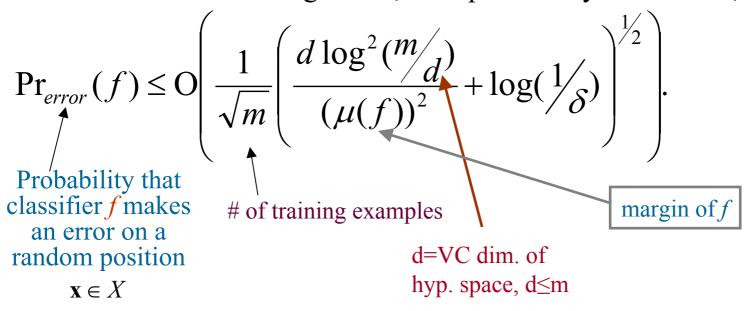
- We want the boosted classifier (defined via λ) to generalize well, i.e., we want it to perform well on data that is not in the training set.
- The margin theory: The margin of a boosted classifier indicates whether it will generalize well. (Schapire et al. '98)
- Large margin classifiers work well in practice, (but there's more to this story).

Generalization Ability of Boosted Classifiers

Can we guess whether a boosted classifier f generalizes well?

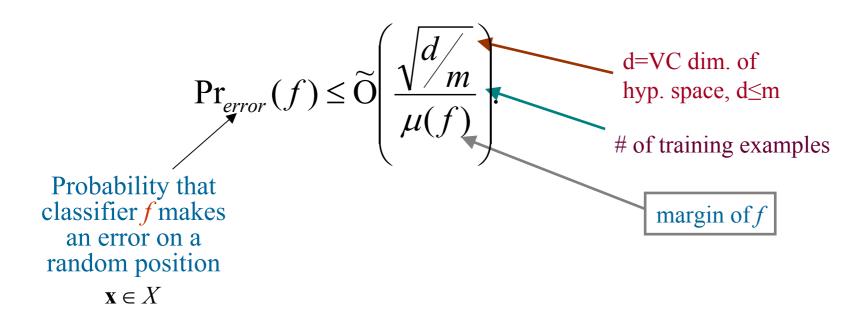
• Can not calculate $Pr_{error}(f)$

Minimize the rhs of a (loose) inequality such as this one (Schapire et al.) When there are no training errors, with probability at least 1- δ ,



The margin theory:

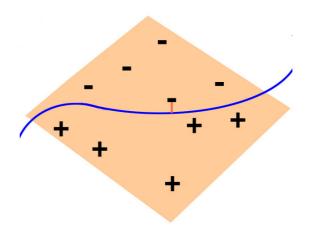
When there are no training errors, with high probability: (Schapire et al, '98)



Large margin = better generalization = smaller probability of error

For Boosting, the margin of combined classifier f_{λ} (where $f_{\lambda} := \text{sign}(\lambda_1 h_1 + ... + \lambda_n h_n)$) is defined by

margin :=
$$\mu(f_{\lambda}) := \min_{i} \frac{(\mathbf{M}\lambda)_{i}}{\|\lambda\|_{1}}$$
.



Does AdaBoost produce maximum margin classifiers?

(AdaBoost was invented before the margin theory...)

(Grove and Schuurmans '98)

- yes, empirically.

(Schapire, et al. '98)

- proved AdaBoost achieves at least half the maximum possible margin.

(Rätsch and Warmuth '03)

- yes, empirically.
- improved the bound.

(R, Daubechies, Schapire '04)

- no, it doesn't.

AdaBoost performs mysteriously well!

AdaBoost performs better than algorithms which are designed to maximize the margin

Still open: Why does AdaBoost work so well?

Does AdaBoost converge?

Better / more predictable boosting algorithms!