Boosting

Choosing a classifier

So many choices:

- Nearest neighbor
- Different generative models
- Linear predictors with different loss functions
- Different kernels
- Neural nets
- etc.

Can one **combine** them?
And get a classifier that is better than any of them individually?

Combining simple classifiers

1. No one classifier is going to be the final product.
   So why not keep the individual components simple?

2. How to train each constituent classifier?
   On the full training set?

3. The full (combined) models may get enormous.
   Is this bad for generalization?

Weak learners

It is often easy to come up with a **weak classifier**, one that is marginally better than random guessing:

$$\Pr(h(X) \neq Y) \leq \frac{1}{2} - \epsilon$$

A learning algorithm that can consistently generate such classifiers is called a **weak learner**.

Is it possible to systematically boost the quality of a weak learner?

The blueprint for boosting

Given: data set $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$.

- Initially give all points equal weight.
- Repeat for $t = 1, 2, \ldots$:
  - Feed weighted data set to the weak learner, get back a weak classifier h_t
  - Reweight data to put more emphasis on points that h_t gets wrong
- Combine all these h_t's linearly

AdaBoost

Data set $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$, labels $y^{(i)} \in \{-1, +1\}$.

1. Initialize $D_1(i) = 1/n$ for all $i = 1, 2, \ldots, n$
2. For $t = 1, 2, \ldots, T$:
   - Give D_t to weak learner, get back some $h_t : \mathcal{X} \to [-1, 1]$
   - Compute h_t's margin of correctness:

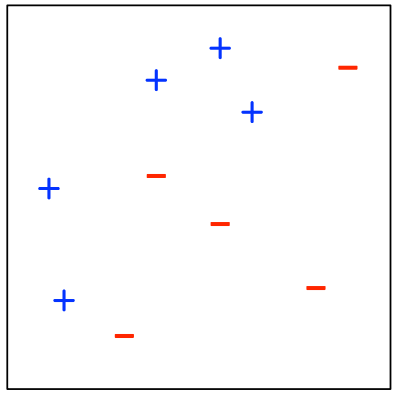
$$r_t = \sum_{i=1}^{n} D_t(i) y^{(i)} h_t(x^{(i)}) \in [-1, 1]$$

$$\alpha_t = \frac{1}{2} \ln \frac{1 + r_t}{1 - r_t}$$

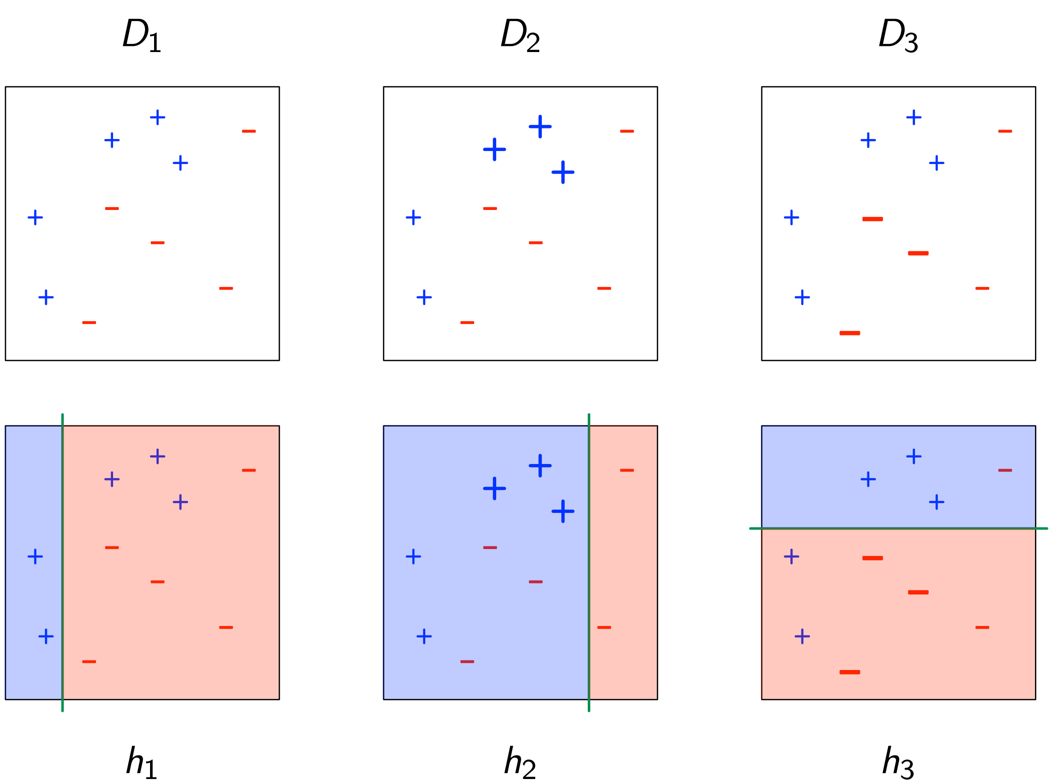
   - Update weights: $D_{t+1}(i) \propto D_t(i) \exp\left(-\alpha_t y^{(i)} h_t(x^{(i)})\right)$
3. Final classifier: $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Example (Freund-Schapire)

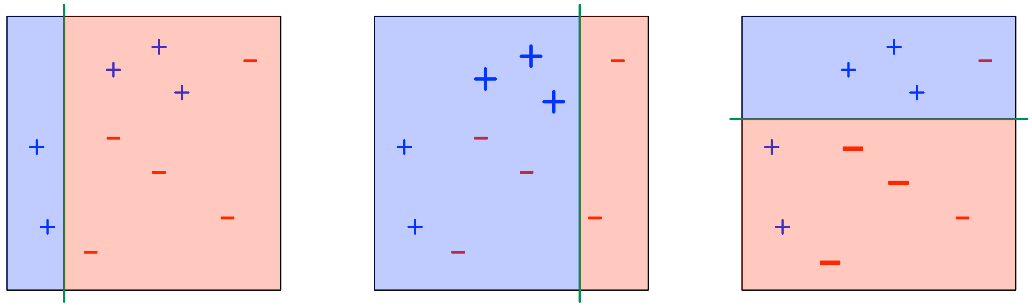
Training set:



Use "decision stumps" (single-feature thresholds) as weak classifiers



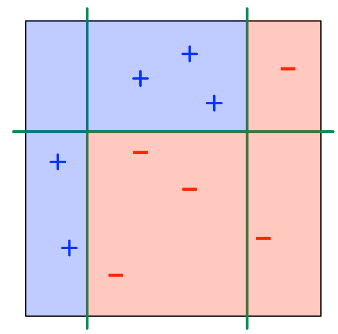
$r_1 = 0.40, \alpha_1 = 0.42$ $r_2 = 0.58, \alpha_2 = 0.65$ $r_3 = 0.72, \alpha_3 = 0.92$



h_1 $\alpha_1 = 0.42$ | h_2 $\alpha_2 = 0.65$ | h_3 $\alpha_3 = 0.92$

Final classifier:

$\text{sign}\left(0.42h_1(x) + 0.65h_2(x) + 0.92h_3(x)\right)$



The surprising power of weak learning

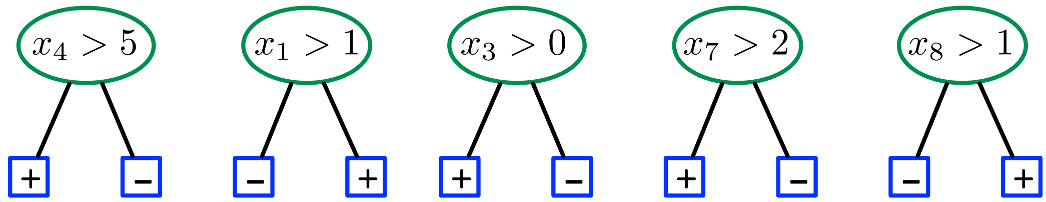
Suppose that on each round t, the weak learner returns a rule h_t whose error on the time-t weighted data distribution is $\leq 1/2 - \gamma$.

Then, after T rounds, the training error of the combined rule

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

is at most $e^{-\gamma^2 T/2}$.

Boosting decision stumps and trees



Boosting decision stumps and trees