Boosting
CSC 411/2515 Tutorial

Wenjie Luo
Nov {24th, 26th, 27th}, 2015

Slides adapted from last year's version by Shenlong Wang

Example: Raptors

- Imagine you are going to bet on Raptors, but don't know much about it.
- What you might do: ask one expert
  - Hard to give a 'decision boundary'
  - Easy to give thumbs up or thumbs down
  - Which expert to ask..
- Question:
  - How to combine/take advantage of suggestions from multiple experts

Ensemble Methods

- Bagging
  - Parallel training with different training sets.

- **Boosting**
  - Sequential training, re-weight training examples to focus on misclassified examples.

- Mixture of experts
  - Parallel training with objective encouraging division of labor.

AdaBoost algorithm

Input:

- Training set examples $\{\boldsymbol{x}_n, t_n\}$, $n = 1,2,\ldots,N$; $t_n \in \{-1,+1\}$
- *WeakLearn*: learning procedure, produces classifier $y(\boldsymbol{x})$

Initialize example weights: $w_n^m(\boldsymbol{x}) = 1/N$

For m=1:M

- $y_m(\boldsymbol{x}) = WeakLearn(\{\boldsymbol{x}\}, \boldsymbol{t}, \boldsymbol{w})$ – fit classifier to training data, minimizing weighted error function:

$$J_m = \sum_{n=1}^{N} w_n^m [y_m(\mathbf{x}_n) \neq t_n]$$

- error rate: $\varepsilon_m = \sum_{n=1}^{N} w_n^m [y_m(\mathbf{x}_n) \neq t_n]$
- classifier coefficient: $\alpha_m = \frac{1}{2} \log\{(1-\varepsilon_m)/\varepsilon_m\}$
- update data weights: $w_n^{m+1} = w_n^m \exp\{-\alpha_m t_n y_m(\mathbf{x}_n)\} / Z^{m+1}$

$$Z^{m+1} = \sum_{n=1}^{N} w_n^m \exp\{-\alpha_m t_n y_m(\mathbf{x}_n)\}$$

Final model:

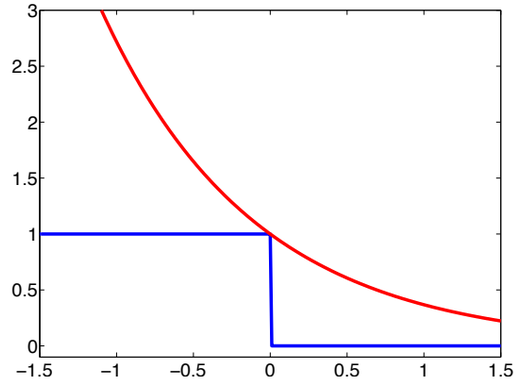
$$Y^M(\mathbf{x}) = \text{sign}(y^M(\mathbf{x})) = \text{sign}(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x}))$$

AdaBoost: Derivation

AdaBoost is equivalent to fitting a greedy forward additive model with this cost function:

$$\text{cost}[t, y(\mathbf{x})] = \exp\{-ty(\mathbf{x})\}$$

At each round of boosting, minimize:



$$\begin{aligned}
E^m &= \sum_{n=1}^{N} \exp\{-\alpha_m t_n y^m(\mathbf{x}_n)\} \\
&= \sum_n \exp\{-\sum_{m'=1}^{m} \alpha_{m'} t_n y_{m'}(\mathbf{x}_n)\} \\
&= \sum_n \exp\{-\sum_{m'=1}^{m-1} \alpha_{m'} t_n y_{m'}(\mathbf{x}_n)\} \exp\{-\alpha_m t_n y_m(\mathbf{x}_n)\} \\
&= \zeta^m \sum_n w_n^m \exp\{-\alpha_m t_n y_m(\mathbf{x}_n)\} \\
&= \zeta^m (\exp\{\alpha_m\} \sum_n w_n^m [y_m(\mathbf{x}_n) \neq t_n] + \exp\{-\alpha_m\} \sum_n w_n^m [y_m(\mathbf{x}_n) = t_n]) \\
&= \zeta^m ((\exp\{\alpha_m\} - \exp\{-\alpha_m\}) \sum_n w_n^m [y_m(\mathbf{x}_n) \neq t_n] + \exp\{-\alpha_m\} \sum_n w_n^m)
\end{aligned}$$

AdaBoost: Derivation (cont.)

From this cost function we derive the updates

$$E_m = \zeta^m \{[\exp(\alpha_m) - \exp(-\alpha_m)] \sum_{n=1}^{N} w_n^m [y_m(\mathbf{x}_n) \neq t_n] + \exp(-\alpha_m) \sum_{n=1}^{N} w_n^m \}$$

Minimize wrt $y_m(\mathbf{x})$: weak learner optimizes, gets error

$$J_m = \sum_{n=1}^{N} w_n^m [y_m(\mathbf{x}_n) \neq t_n]$$

Minimize wrt α_m

Update example weights

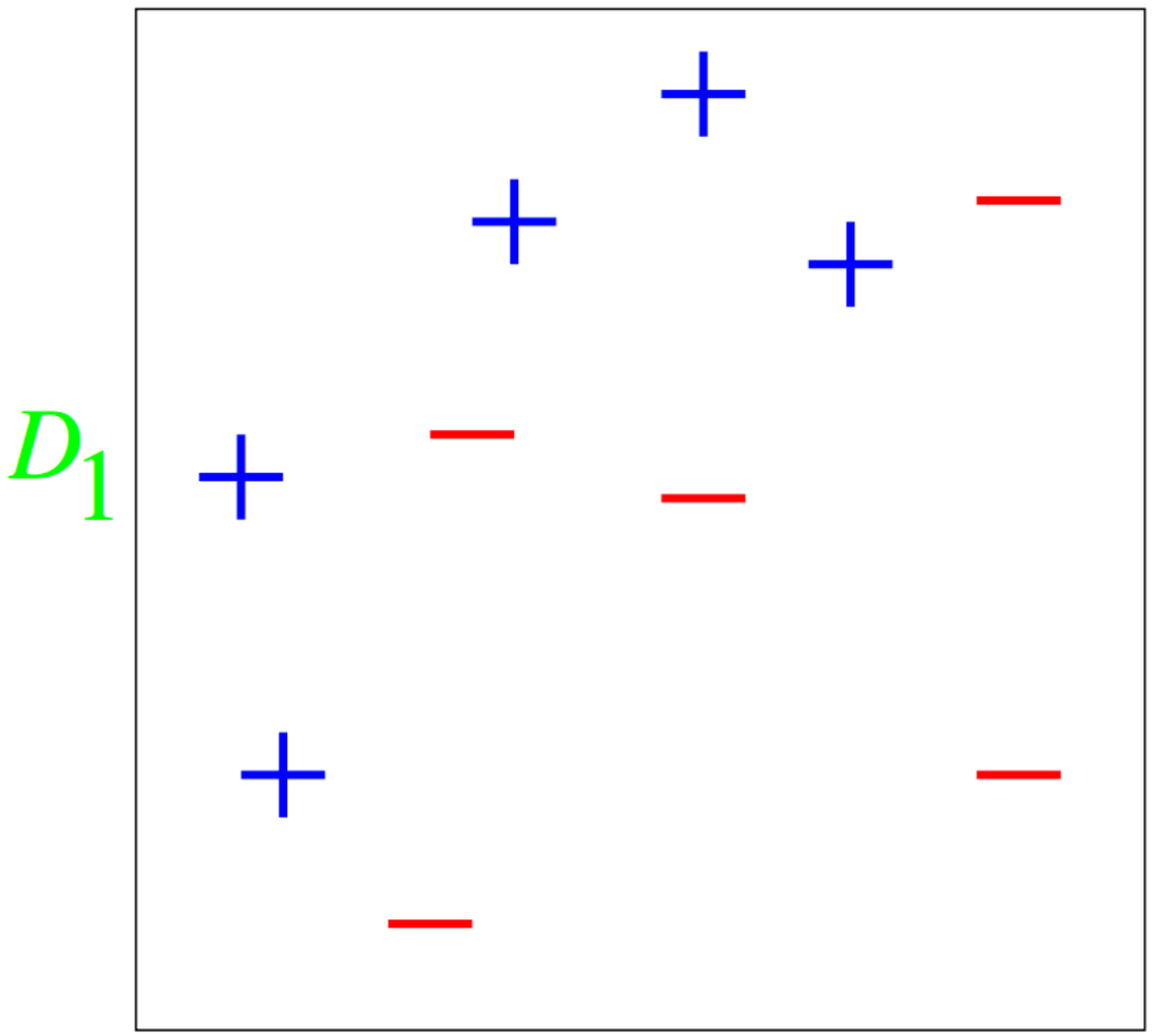
$$w_n^{m+1} \propto \exp\{-\sum_{m'=1}^{m-1} \alpha_{m'} t_n y_{m'}(\mathbf{x}_n)\} \exp\{-\alpha_m t_n y_m(\mathbf{x}_n)\}$$

$$= \zeta^m w_n^m \exp\{-\alpha_m t_n y_m(\mathbf{x}_n)\}$$

$$w_n^{m+1} = w_n^m \exp\{-\alpha_m t_n y_m(\mathbf{x}_n)\} / Z^{m+1}$$

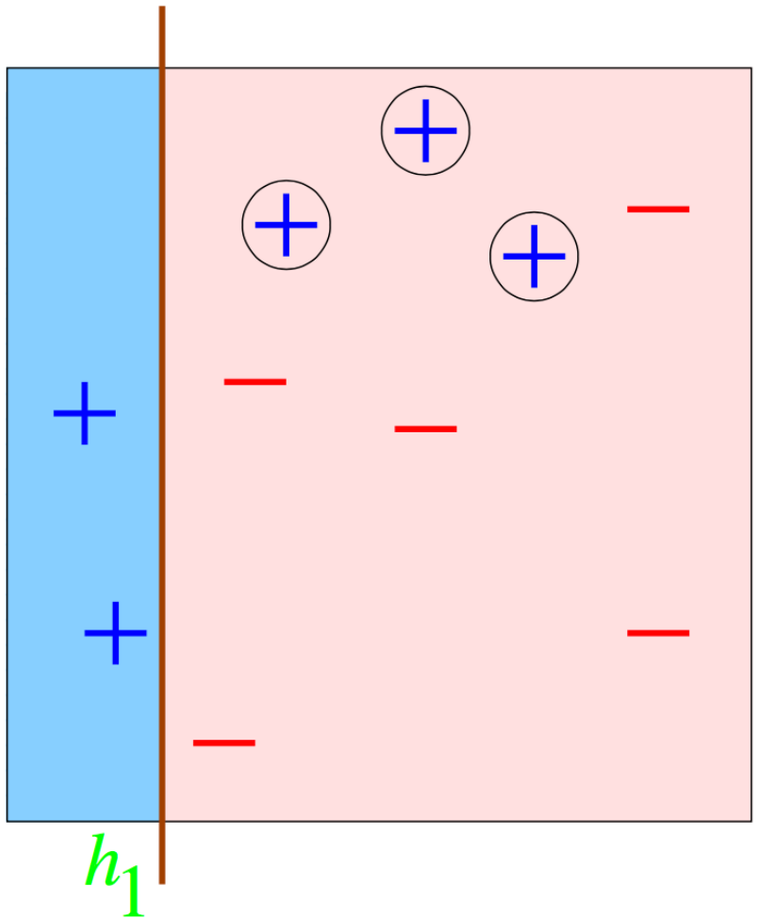
$$Z^{m+1} = \sum_{n=1}^{N} w_n^m \exp\{-t_n \alpha_m y_m(\mathbf{x}_n)\}$$

Toy Example



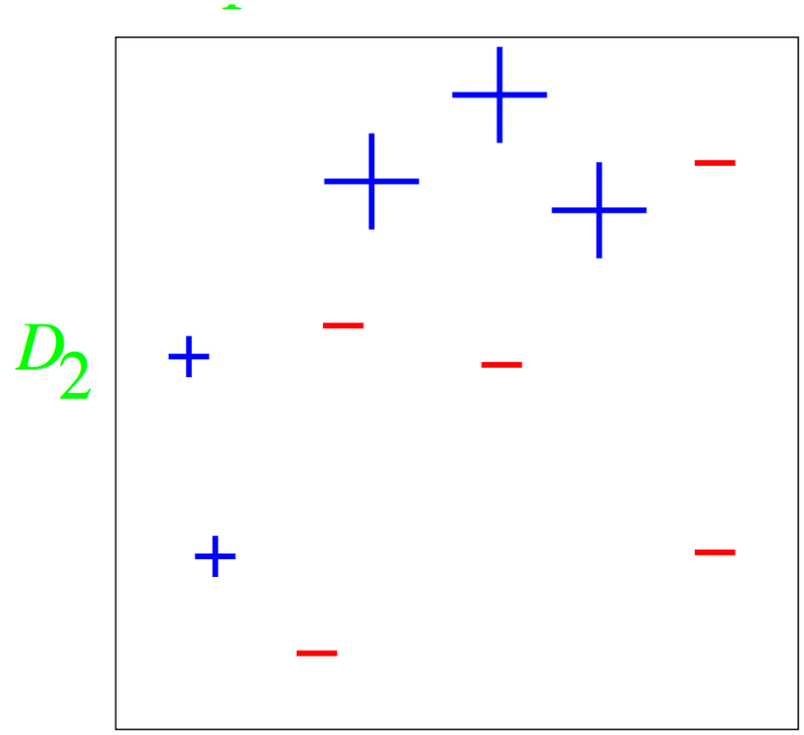
Figures from Yoav Freund and Rob Schapire

Round 1

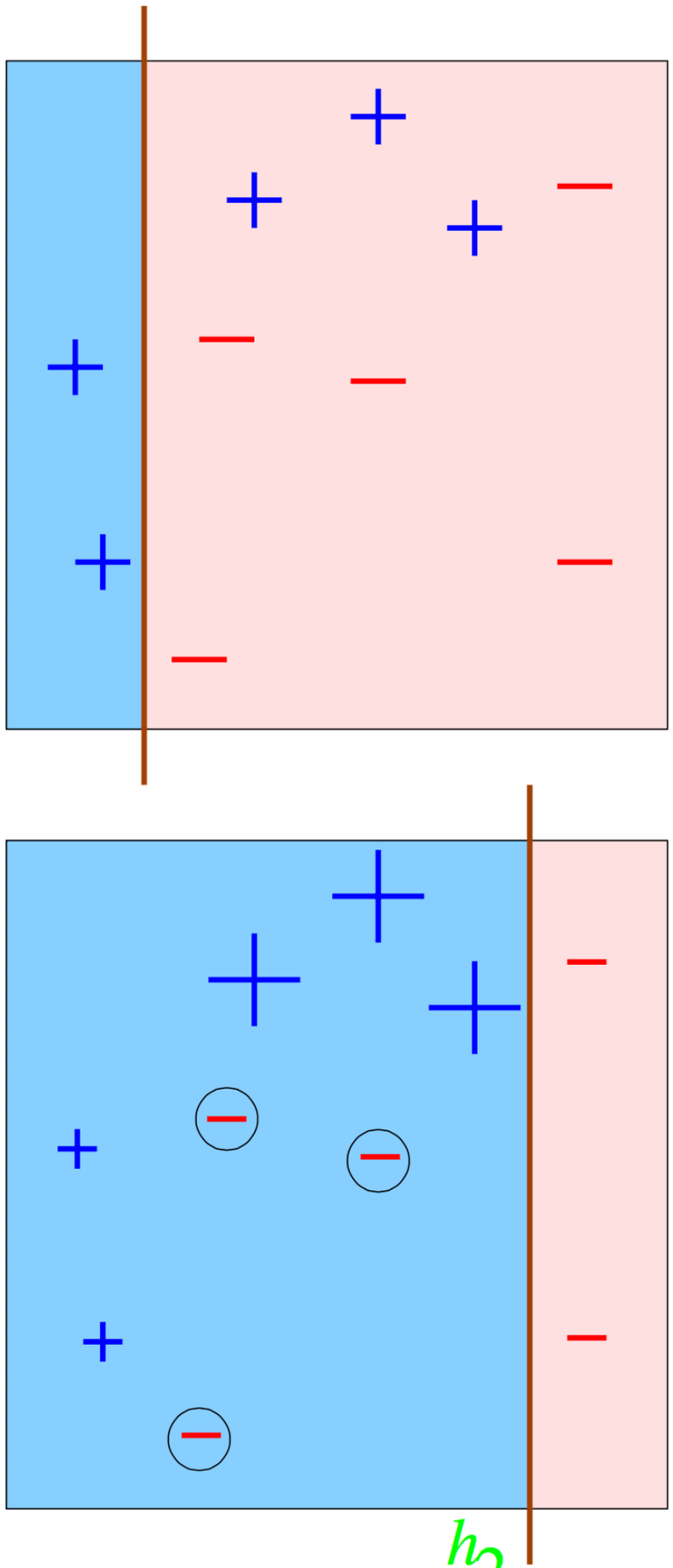


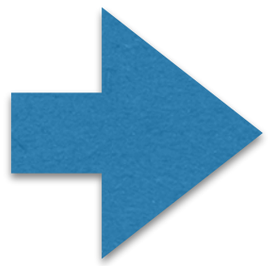
$\varepsilon_1 = 0.30$

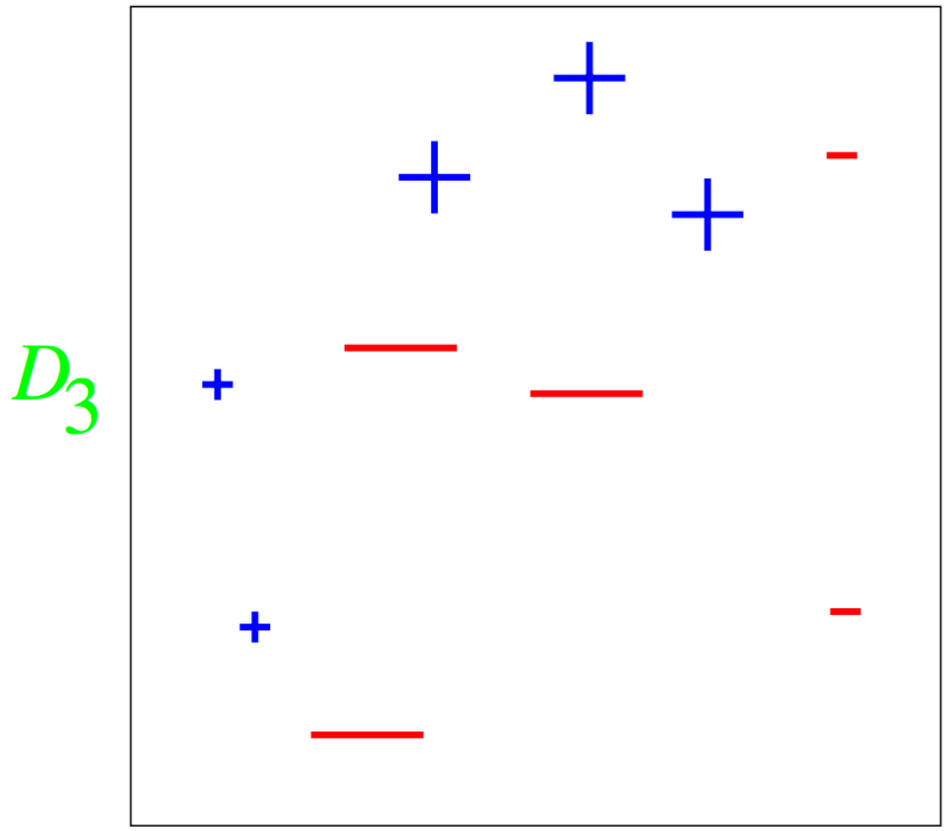
$\alpha_1 = 0.42$



Round 2



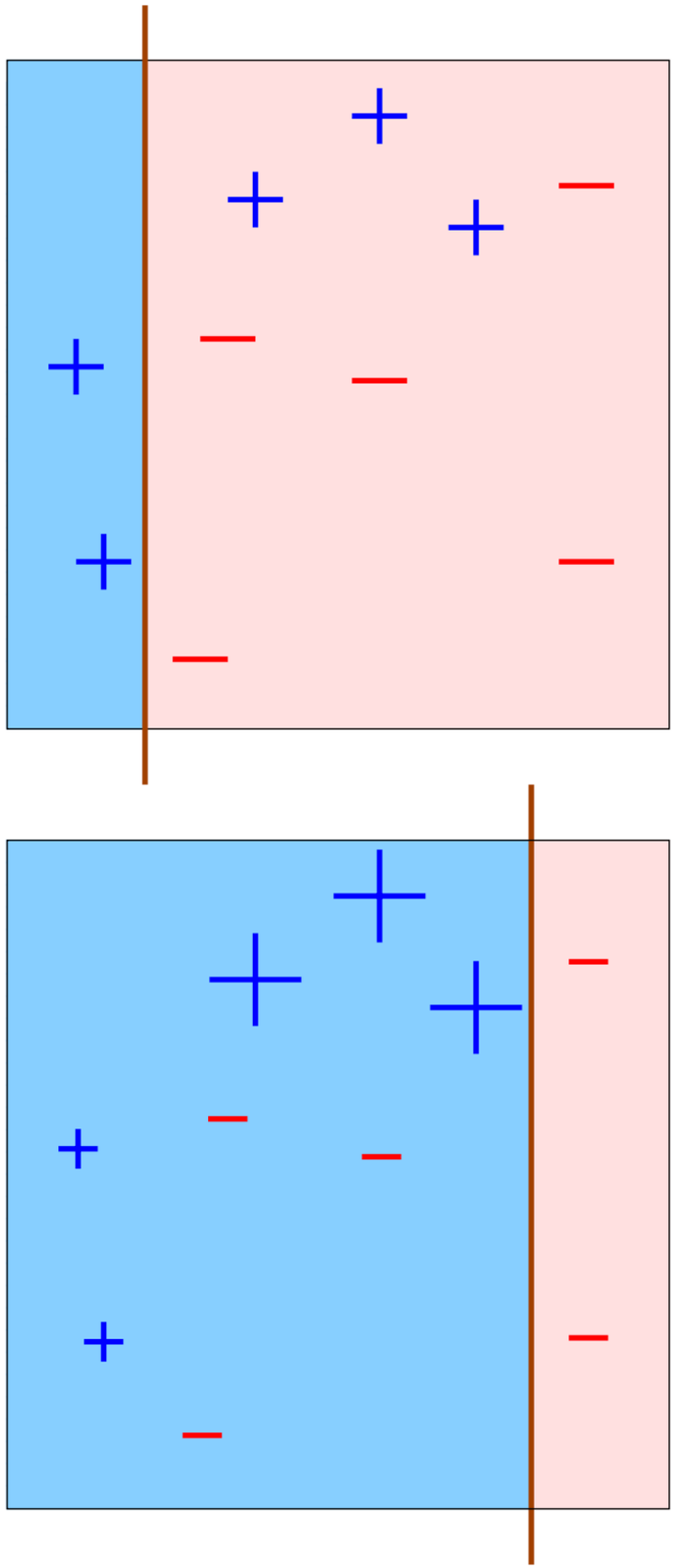


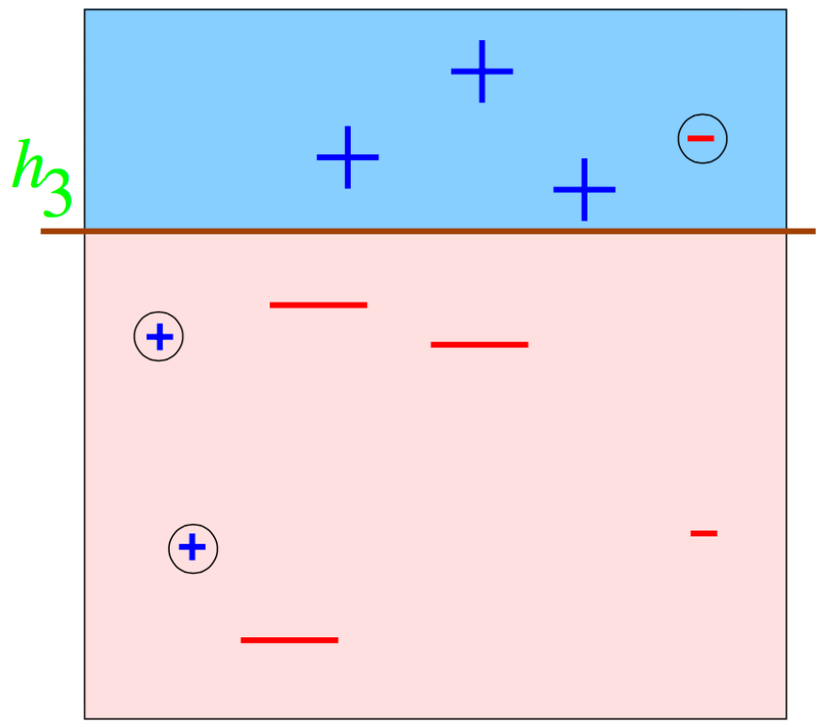


$\varepsilon_2 = 0.21$

$\alpha_2 = 0.65$

Round 3



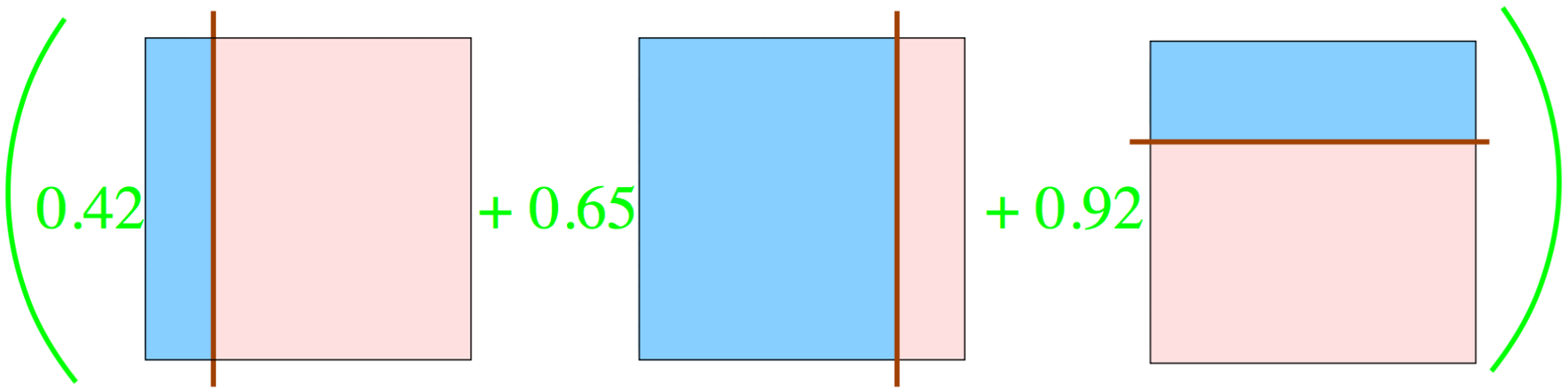


$\varepsilon_3 = 0.14$

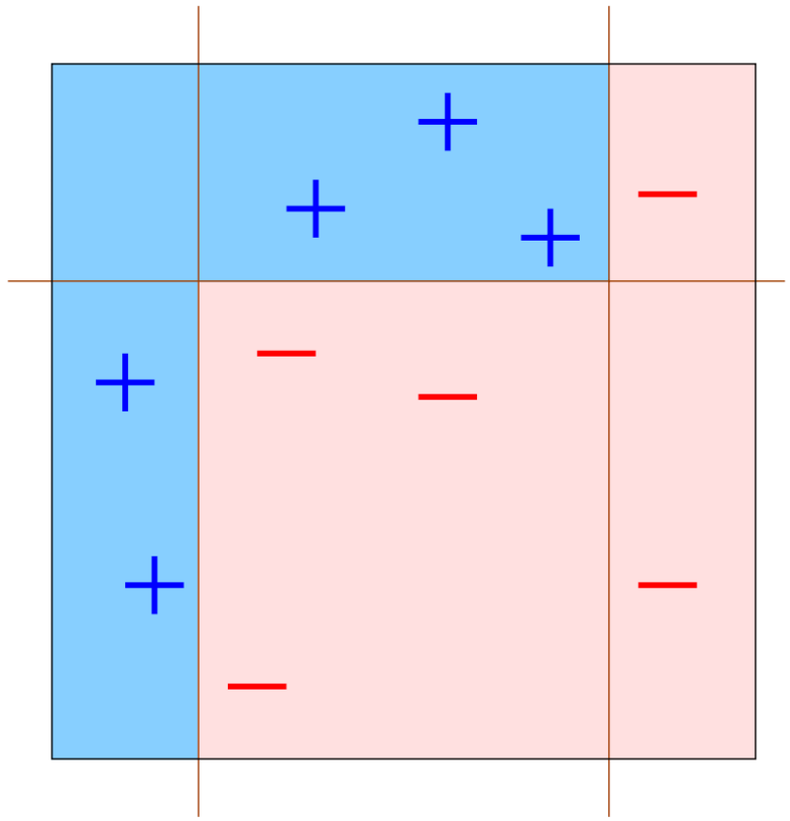
$\alpha_3 = 0.92$

Final Hypothesis

$$H_{\text{final}} = \text{sign}\left(0.42 \; + 0.65 \; + 0.92 \; \right)$$



$=$



AdaBoost

- Adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers.
- Sensitive to noisy data and outliers.
- Primarily reducing bias, and also variance.

Boosting Variations

- Different surrogate loss function
  - E.g. LogitBoost:

    Minimize $\sum_i \log\left(1 + e^{-y_i f(x_i)}\right)$

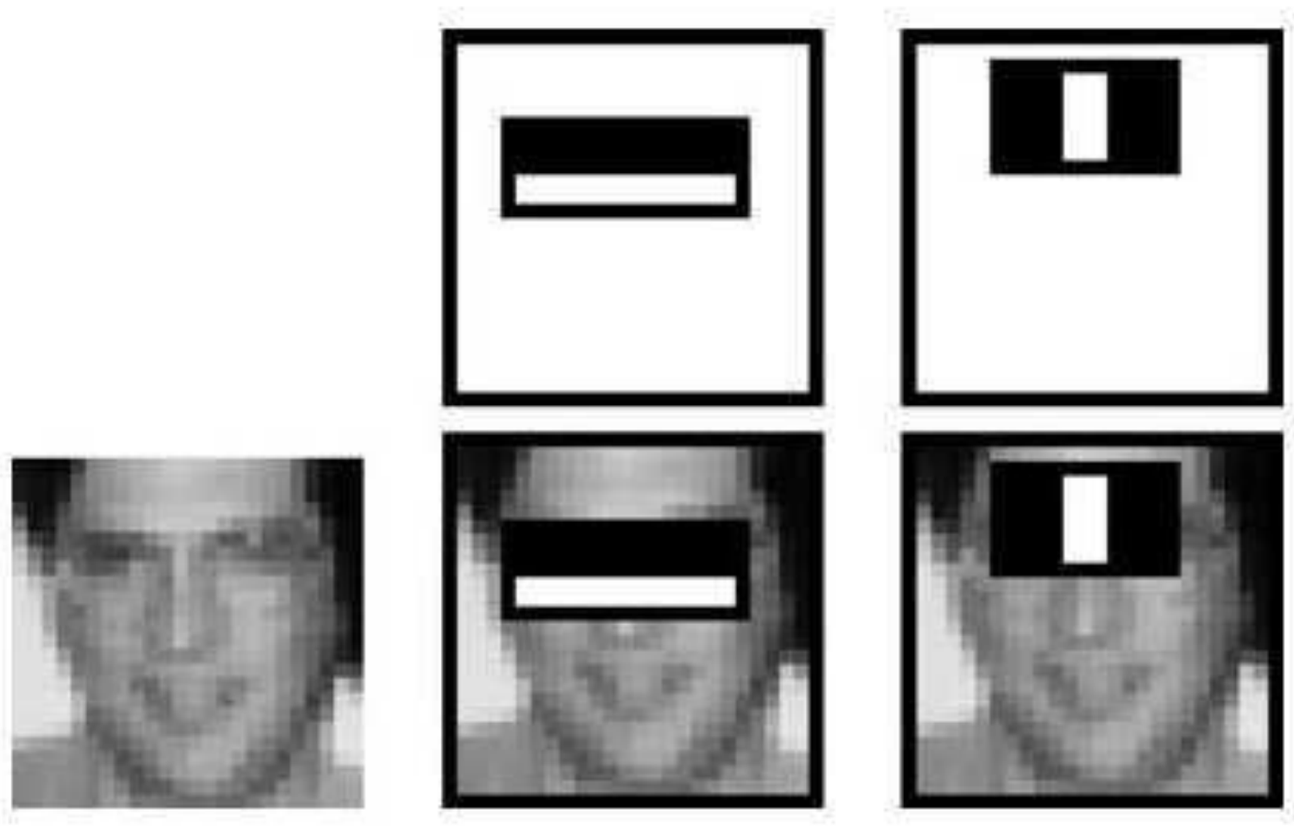
- Removal previous added classifiers
  - E.g. FloatBoost
- Confidence rated version
  - E.g. Realboost

AdaBoost in face detection

Famous application of boosting: detecting faces in images

Two twists on standard algorithm

1) Pre-define weak classifiers, so optimization=selection
2) Change loss function for weak learners: false positives less costly than misses



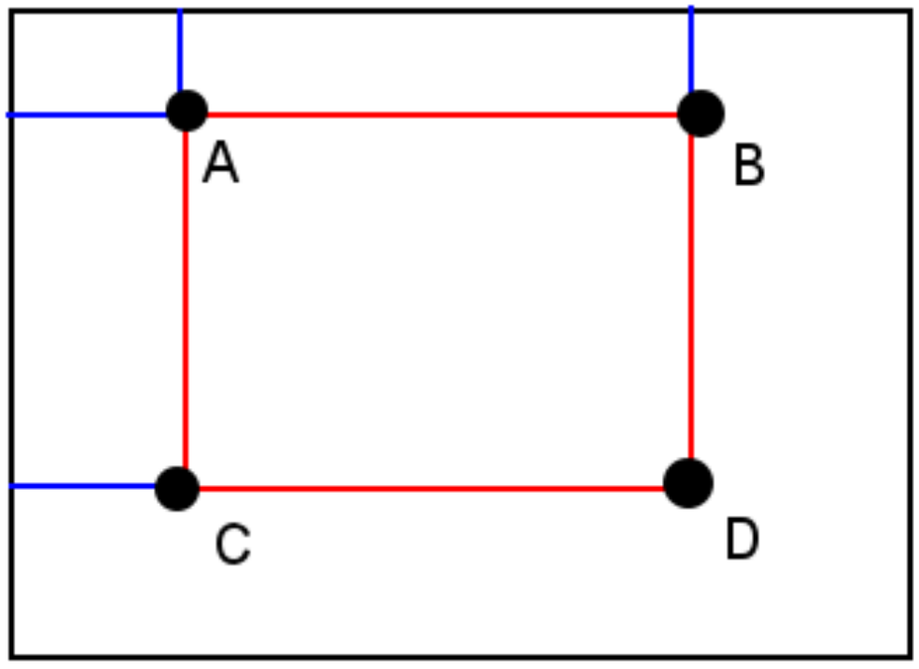
Viola-Jones Detector

Rapid object detection using a boosted cascade of simple features
P **Viola**, M **Jones** - … Vision and Pattern Recognition, 2001. CVPR …, 2001 - ieeexplore.ieee.org
Abstract This paper describes a machine learning approach for visual object detection which is capable of processing images extremely rapidly and achieving high detection rates. This work is distinguished by three key contributions. The first is the introduction of a new **...**
Cited by 10032 Related articles All 125 versions Import into BibTeX Save More

Robust real-time face detection
P Viola, MJ Jones - International journal of computer vision, 2004 - Springer
Abstract This paper describes a **face detection** framework that is capable of processing images extremely rapidly while achieving high **detection** rates. There are three key contributions. The first is the introduction of a new image representation called the " **...**
Cited by 9864 Related articles All 222 versions Import into BibTeX Save More

Integral Image (Summed area table)

$$I(x,y) = \sum_{\substack{x' \le x \\ y' \le y}} i(x',y')$$



Sum = D - B - C + A

$$\sum_{\substack{x0 < x \le x1 \\ y0 < y \le y1}} i(x,y) = I(D) + I(A) - I(B) - I(C).$$