Boosting
(*schema*)

Susana Nascimento

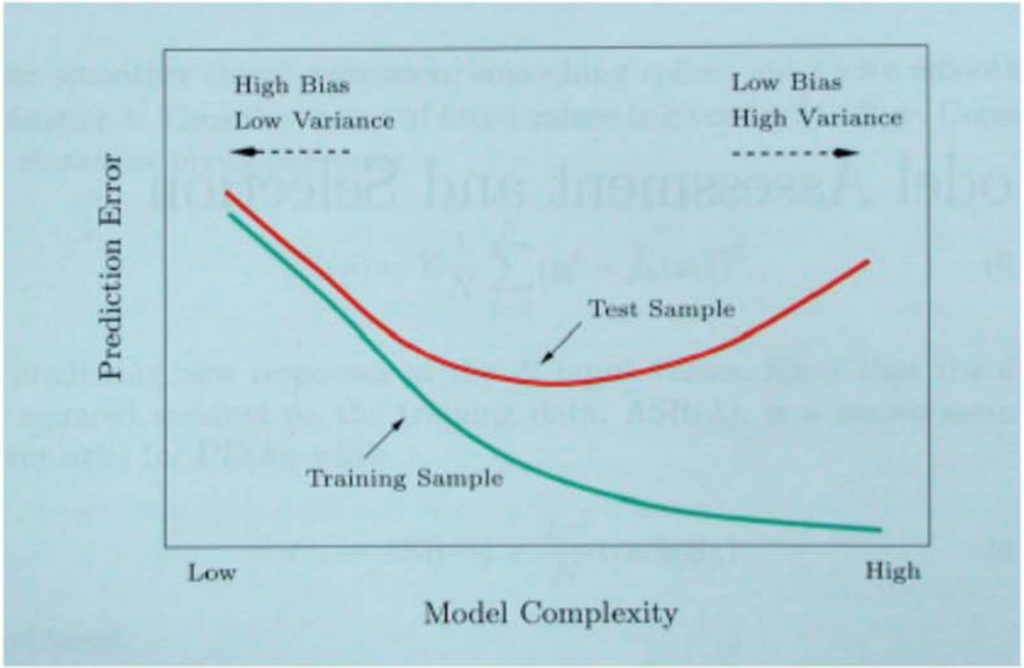
AA – 2016/2017

Last time... Ensemble Methods

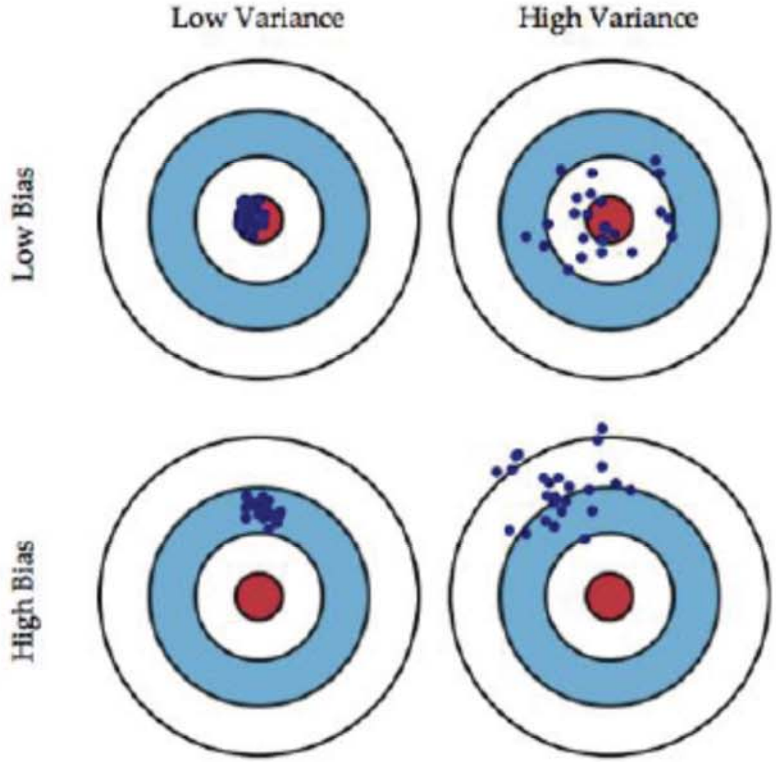
- High level idea
  - Generate multiple hypotheses
  - Combine them to a single classifier

- Two important questions
  - How do we generate multiple hypotheses
    - we have only one sample
  - How do we combine the multiple hypotheses
    - Majority, AdaBoost, ...

Last time… Bias/Variance Tradeoff



Hastie, Tibshirani, Friedman "Elements of Statistical Learning" 2001



Graphical illustration of bias and variance.

http://scott.fortmann-roe.com/docs/BiasVariance.html

Boosting Ideas

- Main idea: use weak learner to create strong learner.
- Ensemble method: combine base classifiers returned by weak learner.
- Finding simple relatively accurate base classifiers often not hard.
- But, how should base classifiers be combined?

Boosting [Schapire, 1989]

- boosting = general method of converting rough rules of thumb into highly accurate prediction rule
- technically:
  - assume given "weak" learning algorithm that can consistently find classifiers ("rules of thumb") at least slightly better than random, say, accuracy $\geq 55\%$ (in two-class setting) ["weak learning assumption"]
  - given sufficient data, a boosting algorithm can provably construct single classifier with very high accuracy, say, 99%

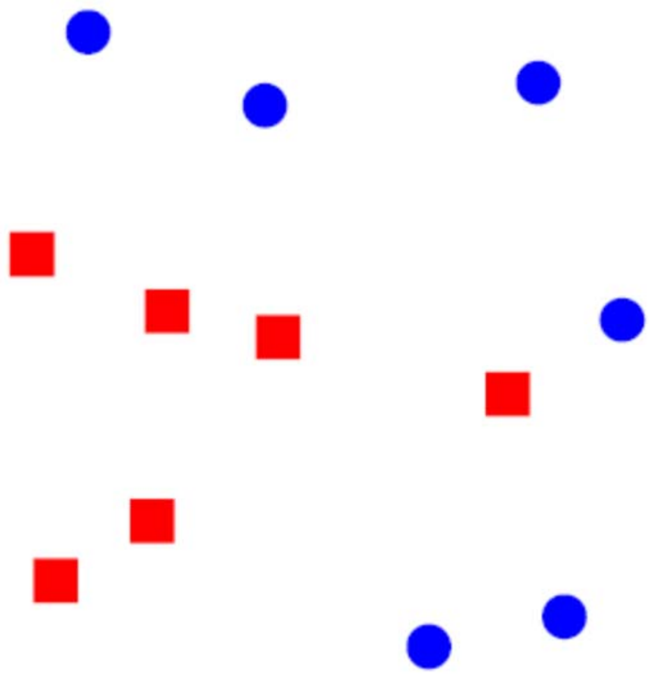
- **Practically useful**
- **Theoretically interesting**

The Boosting Approach

- devise computer program for deriving rough rules of thumb
- apply procedure to subset of examples
- obtain rule of thumb
- apply to 2nd subset of examples
- obtain 2nd rule of thumb
- repeat T times

Boosting: Intuition

- Want to pick weak classifiers that contribute something to the ensemble



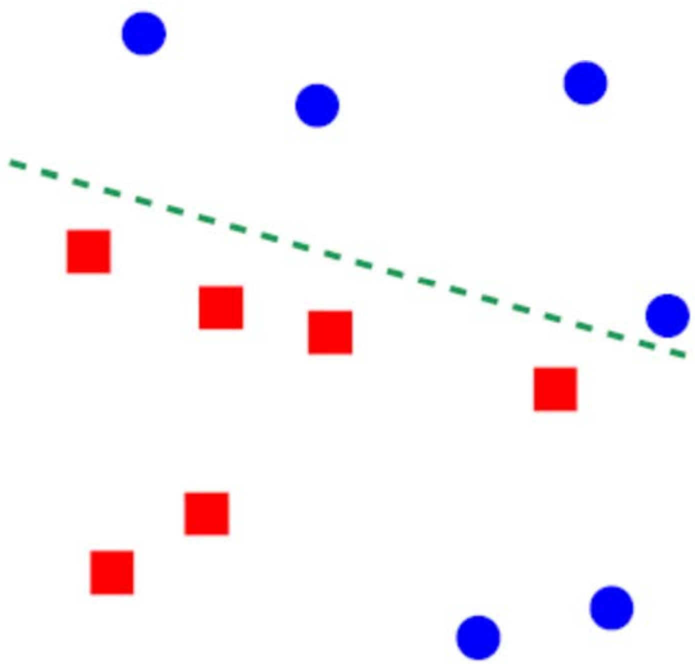
Greedy algorithm: for $m=1,...,M$

- Pick a weak classifier h_m
- Adjust weights: misclassified examples get "heavier"
- α_m set according to weighted error of h_m

[Source: G. Shakhnarovich]

Boosting: Intuition

- Want to pick weak classifiers that contribute something to the ensemble

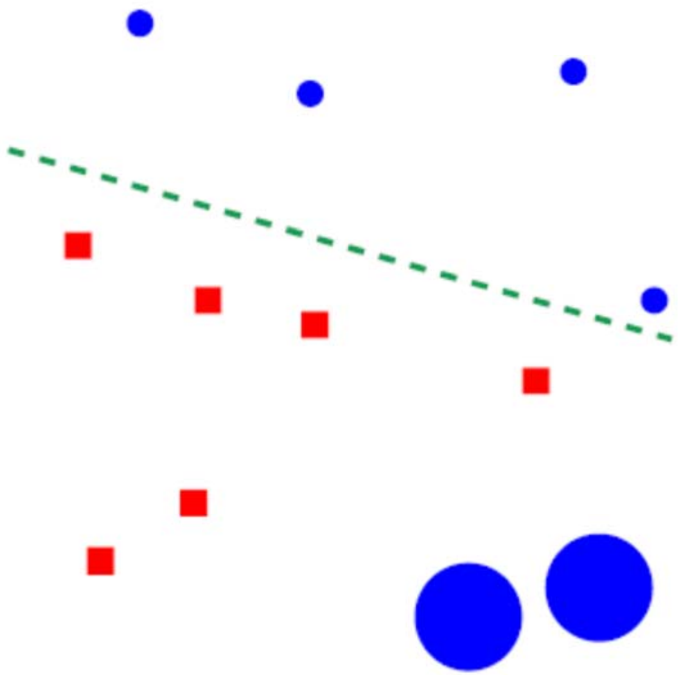


Greedy algorithm: for $m=1,\ldots,M$

- Pick a weak classifier h_m
- Adjust weights: misclassified examples get "heavier"
- α_m set according to weighted error of h_m

Boosting: Intuition

- Want to pick weak classifiers that contribute something to the ensemble

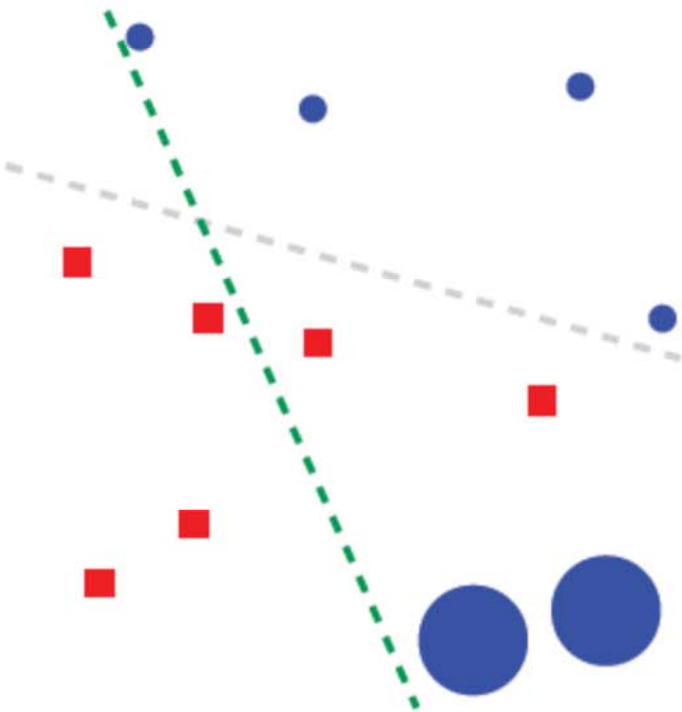


Greedy algorithm: for m=1,...,M

- Pick a weak classifier h_m
- Adjust weights: misclassified examples get "heavier"
- α_m set according to weighted error of h_m

Boosting: Intuition

- Want to pick weak classifiers that contribute something to the ensemble

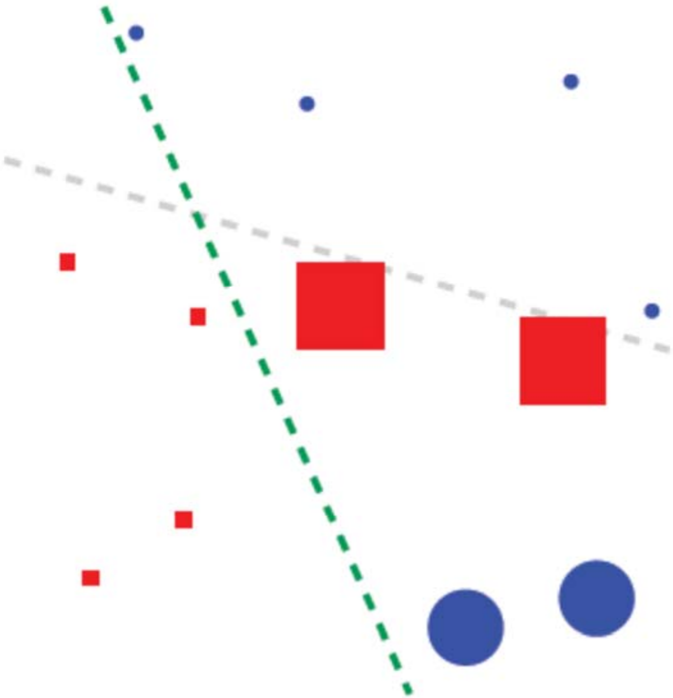


Greedy algorithm: for $m=1,\ldots,M$

- Pick a weak classifier h_m
- Adjust weights: misclassified examples get "heavier"
- α_m set according to weighted error of h_m

Boosting: Intuition

- Want to pick weak classifiers that contribute something to the ensemble

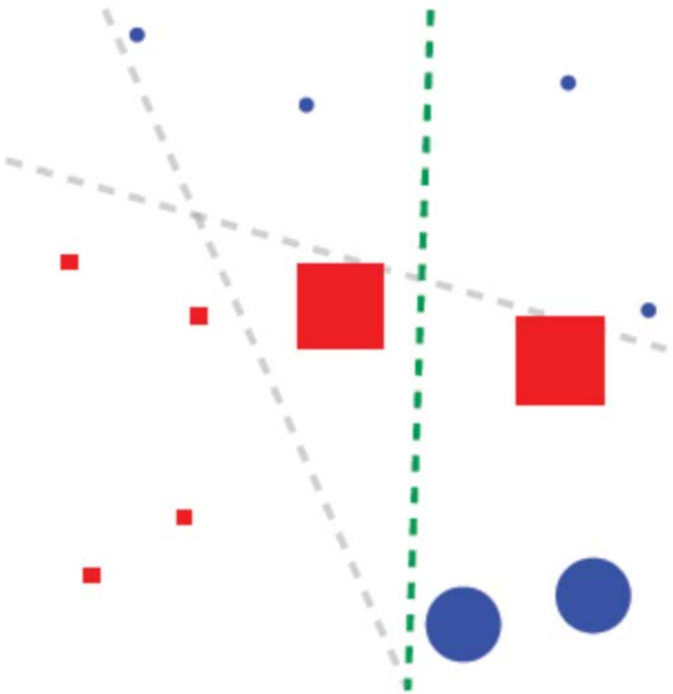


Greedy algorithm: for m=1,...,M

- Pick a weak classifier h_m
- Adjust weights: misclassified examples get "heavier"
- α_m set according to weighted error of h_m

Boosting: Intuition

- Want to pick weak classifiers that contribute something to the ensemble

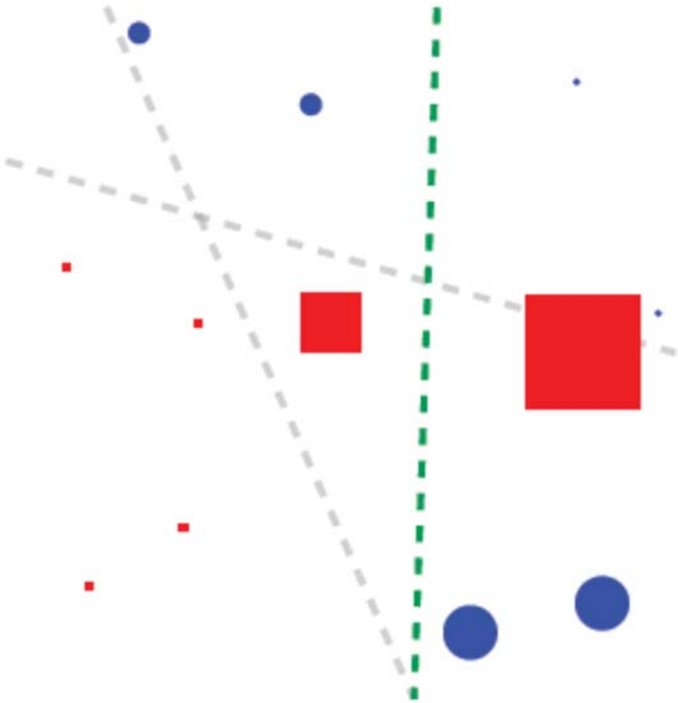


Greedy algorithm: for m=1,...,M

- Pick a weak classifier h_m
- Adjust weights: misclassified examples get "heavier"
- α_m set according to weighted error of h_m

Boosting: Intuition

- Want to pick weak classifiers that contribute something to the ensemble



Greedy algorithm: for $m=1,\ldots,M$

- Pick a weak classifier h_m
- Adjust weights: misclassified examples get "heavier"
- α_m set according to weighted error of h_m

Formalizing Boosting

- given training set $(x_1, y_1), \ldots, (x_m, y_m)$
- $y_i \in \{-1, +1\}$ correct label of instance $x_i \in X$
- for $t = 1, \ldots, T$:
  - construct distribution D_t on $\{1, \ldots, m\}$
  - find weak classifier ("rule of thumb")

    $$h_t : X \to \{-1, +1\}$$

    with error ϵ_t on D_t:

    $$\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$$
- output final/combined classifier H_{final}

AdaBoost Algorithm [Freund & Schapire '95]

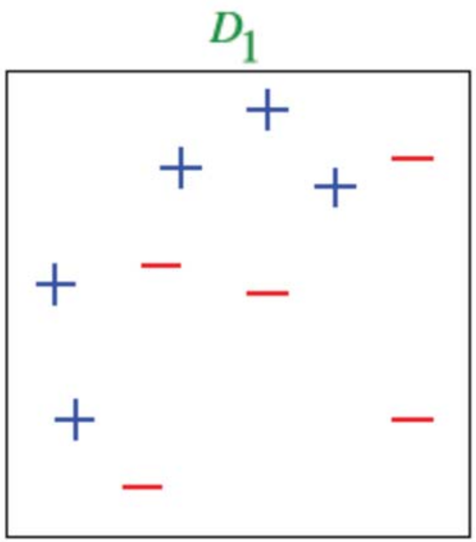
- constructing D_t:
  - $D_1(i) = 1/m$
  - given D_t and h_t:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$
$$= \frac{D_t(i)}{Z_t} \exp(-\alpha_t \, y_i \, h_t(x_i))$$

where $Z_t =$ normalization factor
$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) > 0$$

- final classifier:
  - $H_{\text{final}}(x) = \text{sign}\left(\sum_t \alpha_t h_t(x)\right)$

Toy Example

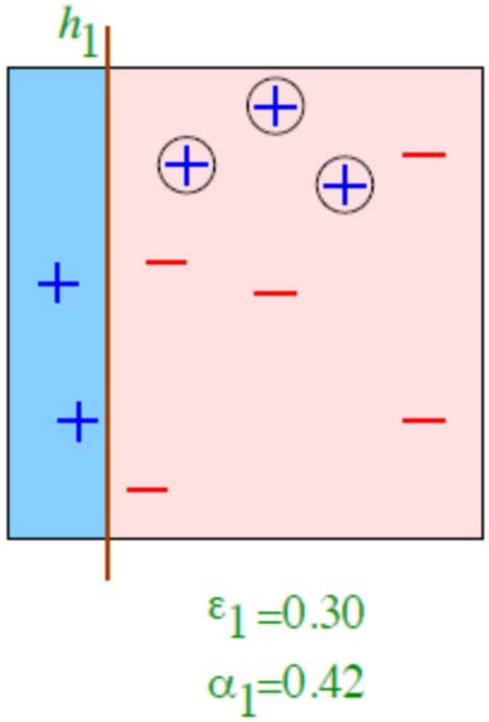


Minimize the error

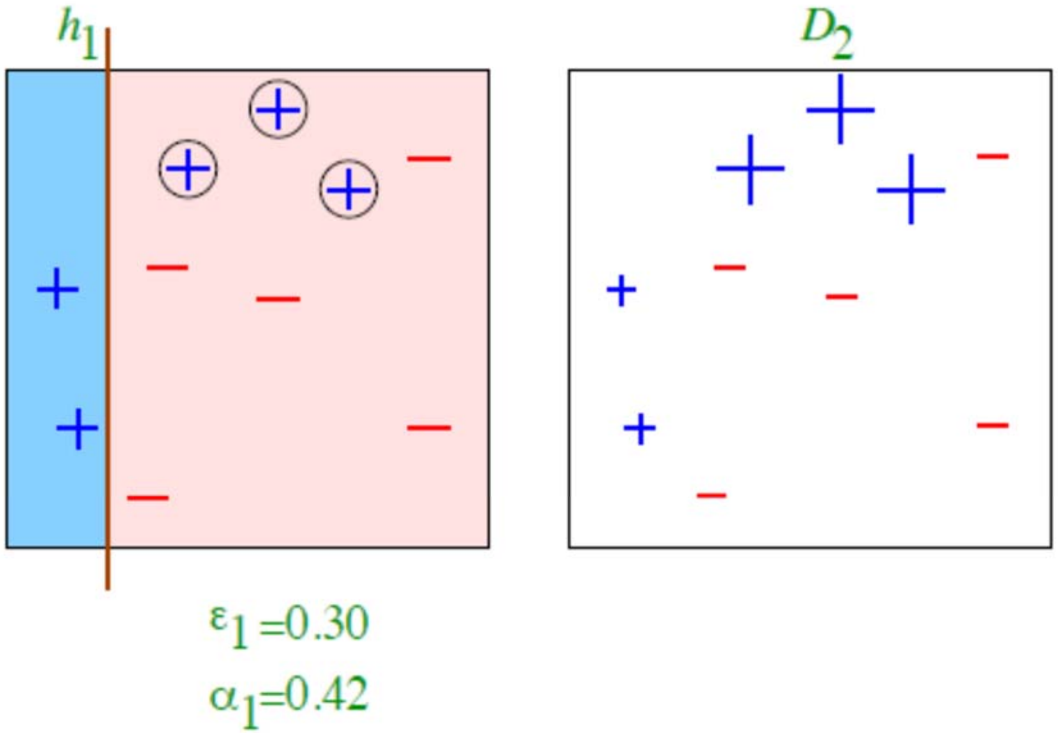
$$\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$$

- Weak hypotheses: vertical or horizontal half-planes

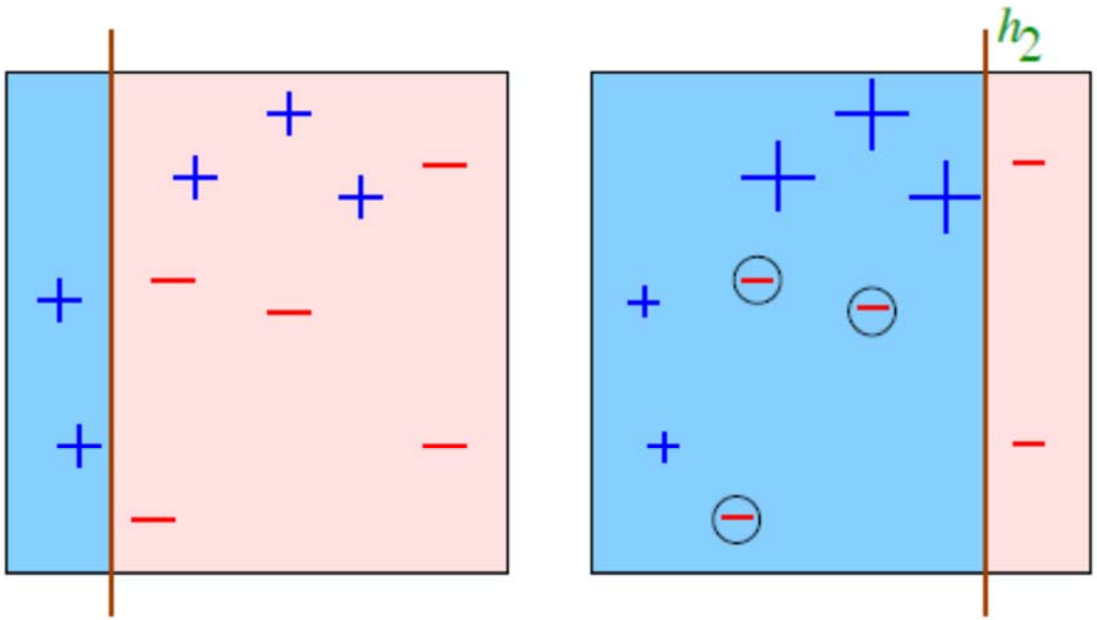
Round 1



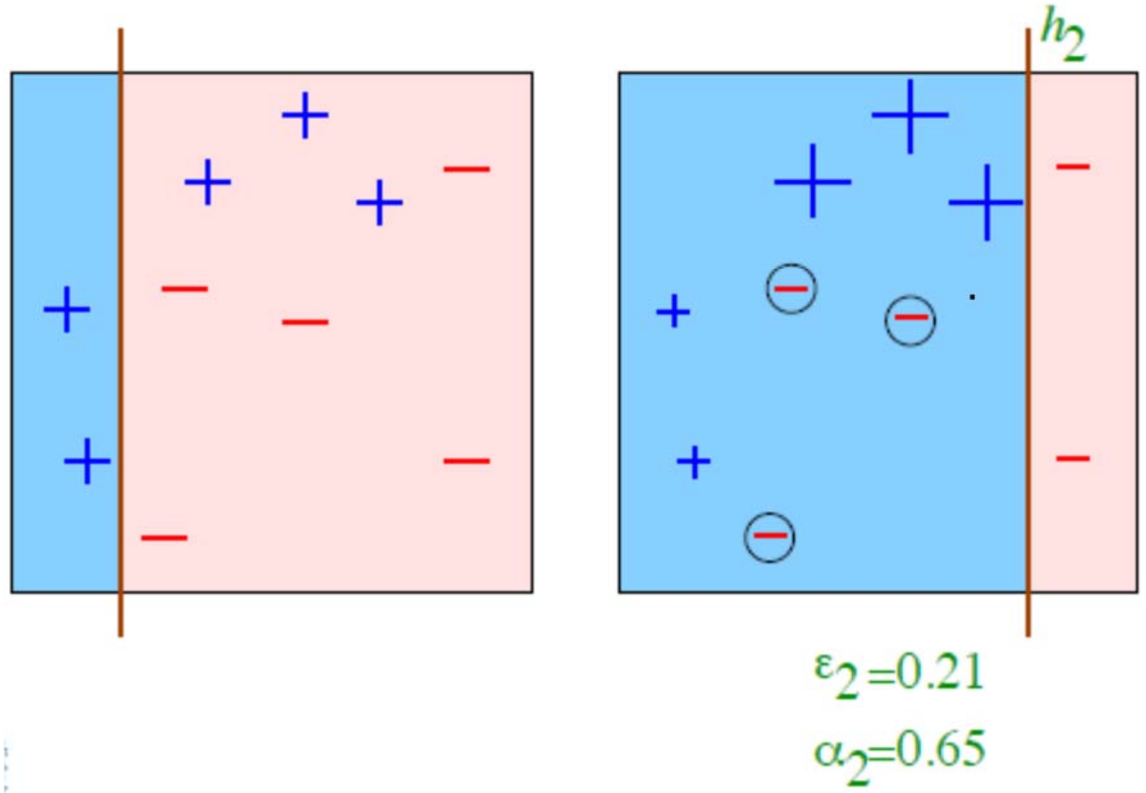
Round 1



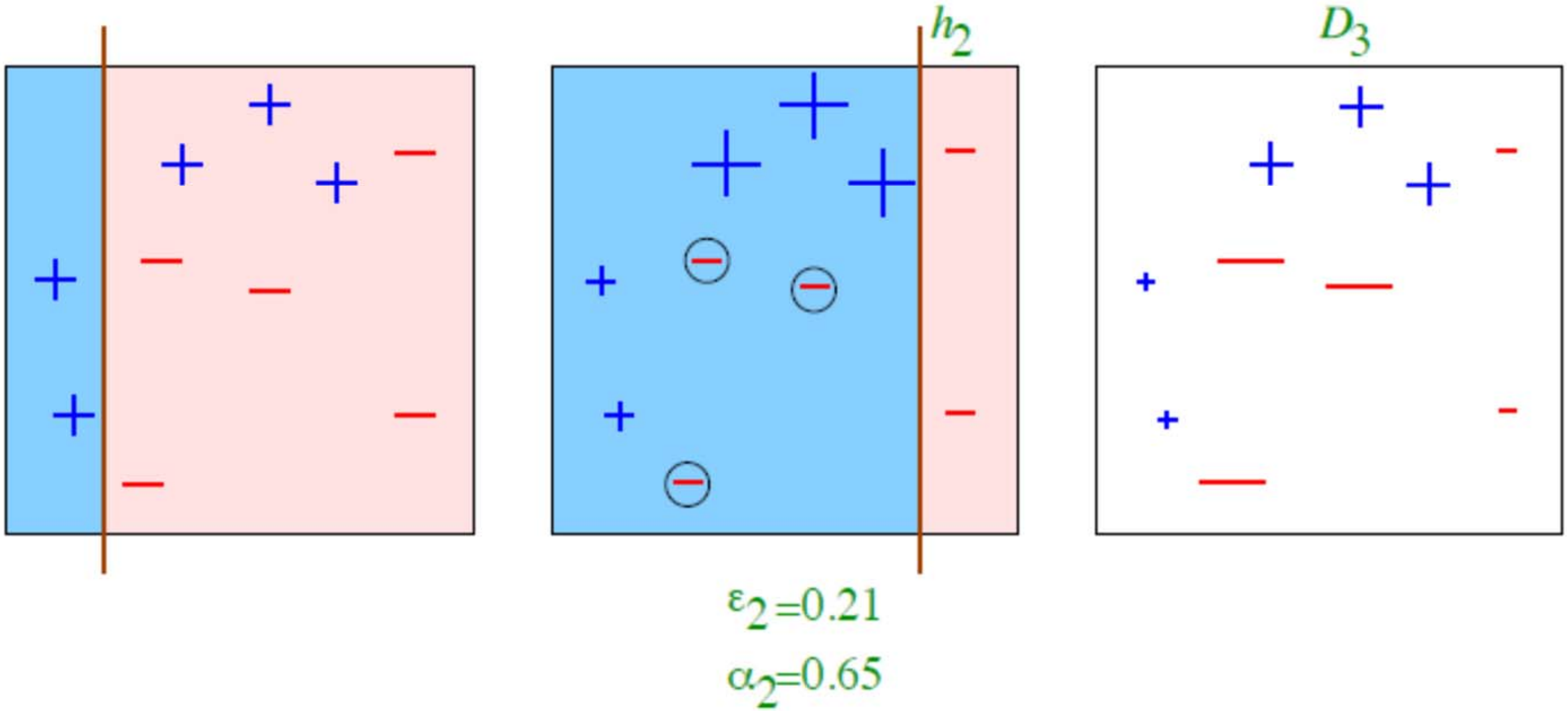
Round 2



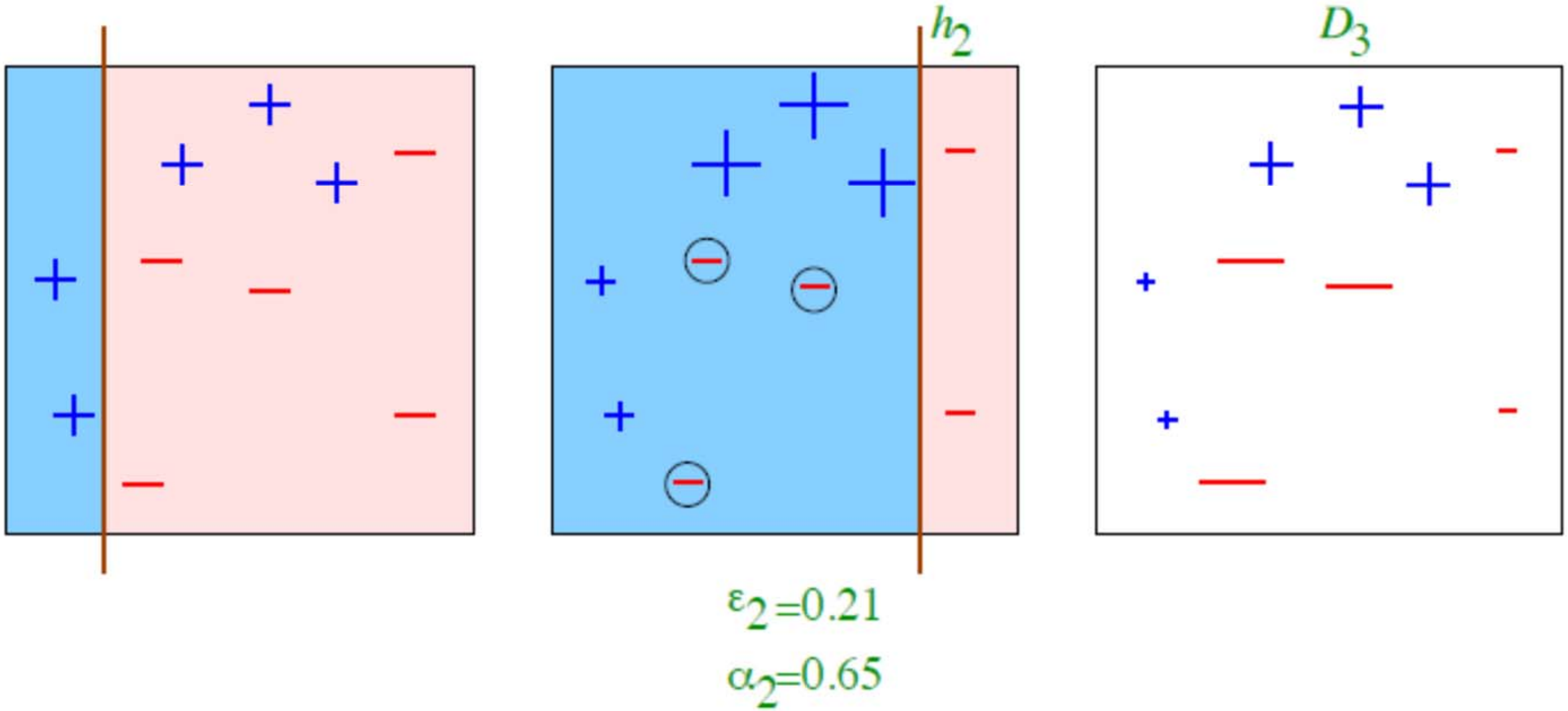
Round 2



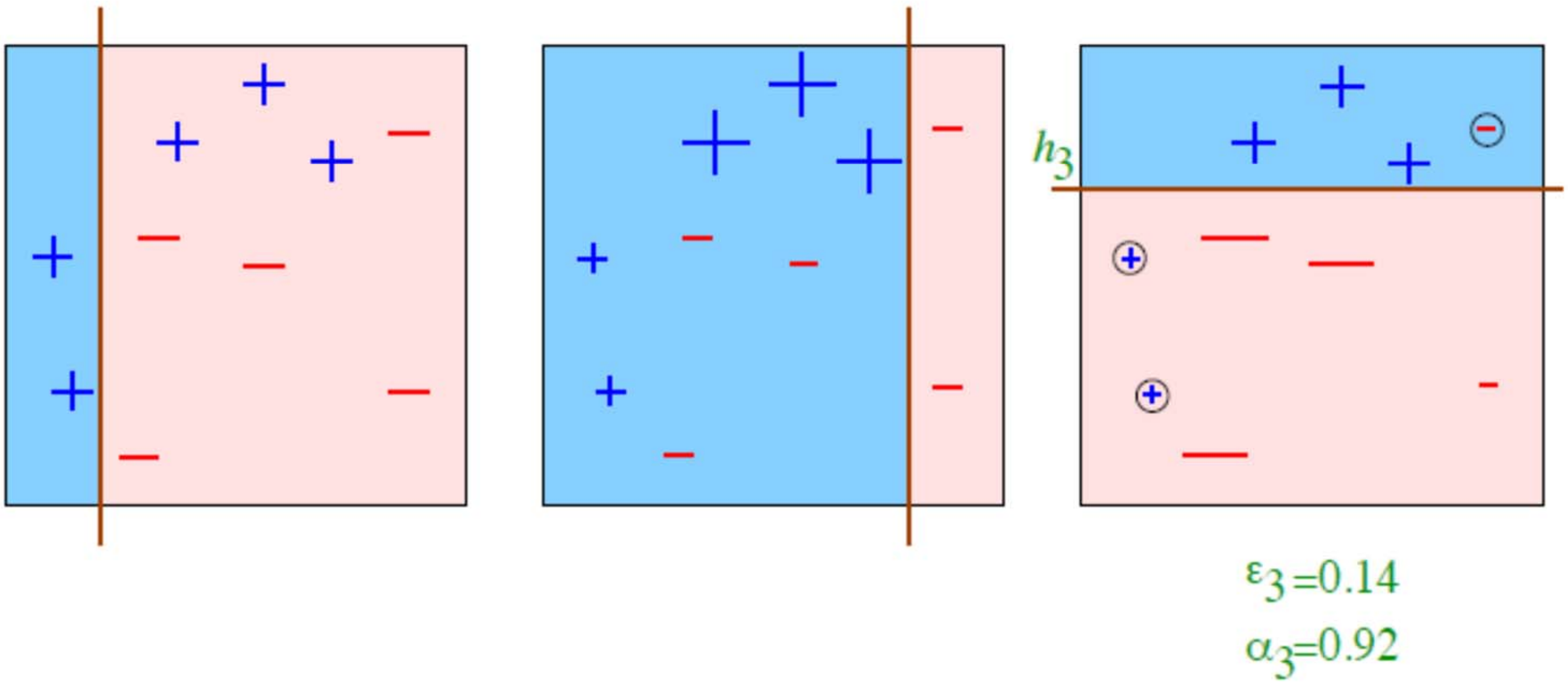
Round 2



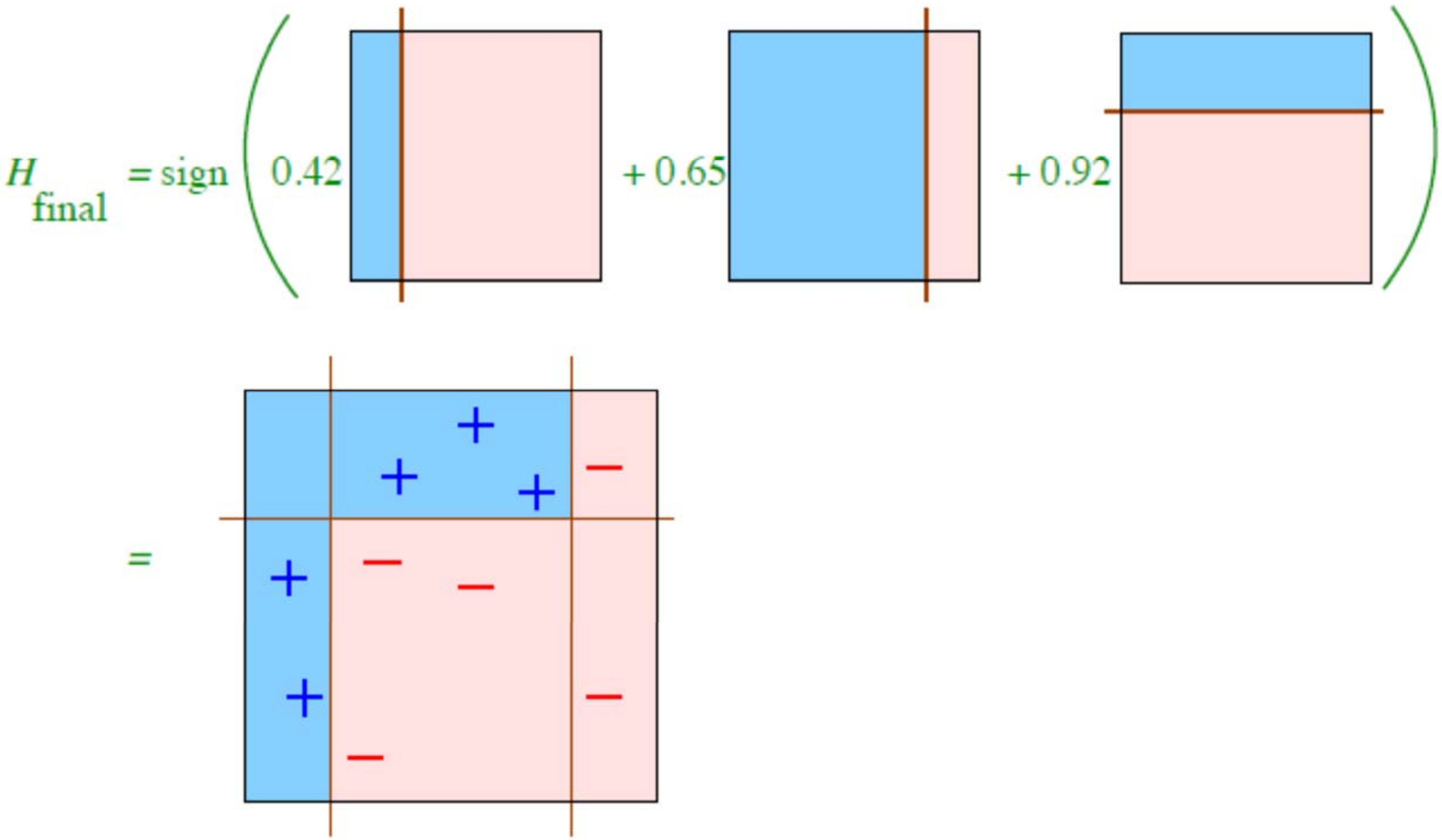
Round 3



Round 3



Final Hypothesis



Boosting *vs* Bagging

Bagging:

- Resample data points
- Weight of each classifier is the same
- Only variance reduction

Boosting:

- Reweights data points (modifies their distribution)
- Weight is dependent on classifier's accuracy
- Both bias and variance reduced – learning rule becomes more complex with iterations