ITCS 6156/8156 Fall 2023
Machine Learning

Ensemble Methods

Instructor: Hongfei Xue
Email: hongfei.xue@charlotte.edu
Class Meeting: Mon & Wed, 4:00 PM – 5:15 PM, CHHS 376

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

Some content in the slides is based on Dr. Raquel Urtasun's lecture

Boosting (AdaBoost)

- Also works by manipulating training set, but classifiers trained sequentially
- Each classifier trained given knowledge of the performance of previously trained classifiers: focus on hard examples
- Final classifier: weighted sum of component classifiers

Making Weak Learners Stronger

- Suppose you have a weak learning module (a base classifier) that can always get $(0.5 + \epsilon)$ correct when given a two-way classification task
  - That seems like a weak assumption but beware!
- Can you apply this learning module many times to get a strong learner that can get close to zero error rate on the training data?
  - Theorists showed how to do this and it actually led to an effective new learning procedure (Freund & Shapire, 1996)

Boosting (AdaBoost)

- First train the base classifier on all the training data with equal importance weights on each case.
- Then re-weight the training data to emphasize the hard cases and train a second model.
  - How do we re-weight the data?
- Keep training new models on re-weighted data
- Finally, use a weighted committee of all the models for the test data.
  - How do we weight the models in the committee?

Train Each Classifier

- Input: $\mathbf{x}$, Output: $y(\mathbf{x}) \in \{1, -1\}$
- Target $t \in \{-1, 1\}$
- Weight on example n for classifier m: $\mathbf{w}_n^m$
- Cost function for classifier m

$$J_m = \sum_{n=1}^{N} w_n^m \underbrace{[y_m(\mathbf{x}^n) \neq t^{(n)}]}_{1 \text{ if error}, \quad 0 \text{ o.w.}} = \sum \text{weighted errors}$$

Weight each training case for classifier m

- Recall cost function is

$$J_m = \sum_{n=1}^{N} w_n^m \underbrace{[y_m(\mathbf{x}^n) \neq t^{(n)}]}_{1 \text{ if error}, \quad 0 \text{ o.w.}} = \sum \text{weighted errors}$$

- Weighted error rate of a classifier

$$\epsilon_m = \frac{J_m}{\sum w_n^m}$$

- The quality of the classifier is

$$\alpha_m = \ln\left(\frac{1-\epsilon_m}{\epsilon_m}\right)$$

  It is zero if the classifier has weighted error rate of 0.5 and infinity if the classifier is perfect

- The weights for the next round are then

$$w_n^{m+1} = \exp\left(-\frac{1}{2}t^{(n)}\sum_{i=1}^{m}\alpha_i y_i(\mathbf{x}^{(n)})\right) = w_n^m \exp\left(-\frac{1}{2}t^{(n)}\alpha_m y_m(\mathbf{x}^{(n)})\right)$$

Make predictions using a committee of classifiers

- Weight the binary prediction of each classifier by the quality of that classifier:

$$y_M(\mathbf{x}) = \text{sign}\left(\sum_{m=1}^{M} \frac{1}{2}\alpha_m y_m(\mathbf{x})\right)$$

- This is how to do inference, i.e., how to compute the prediction for each new example.

AdaBoost Algorithm

- Input: $\{\mathbf{x}^{(n)}, t^{(n)}\}_{n=1}^{N}$, and WeakLearn: learning procedure, produces classifier $y(\mathbf{x})$
- Initialize example weights: $w_n^m(\mathbf{x}) = 1/N$
- For m=1:M
  - $y_m(\mathbf{x}) = WeakLearn(\{\mathbf{x}\}, \mathbf{t}, \mathbf{w})$, fit classifier by minimizing

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

  - Compute unnormalized error rate

$$\epsilon_m = \frac{J_m}{\sum w_n^m}$$

  - Compute classifier coefficient $\alpha_m = \log \frac{1-\epsilon_m}{\epsilon_m}$
  - Update data weights

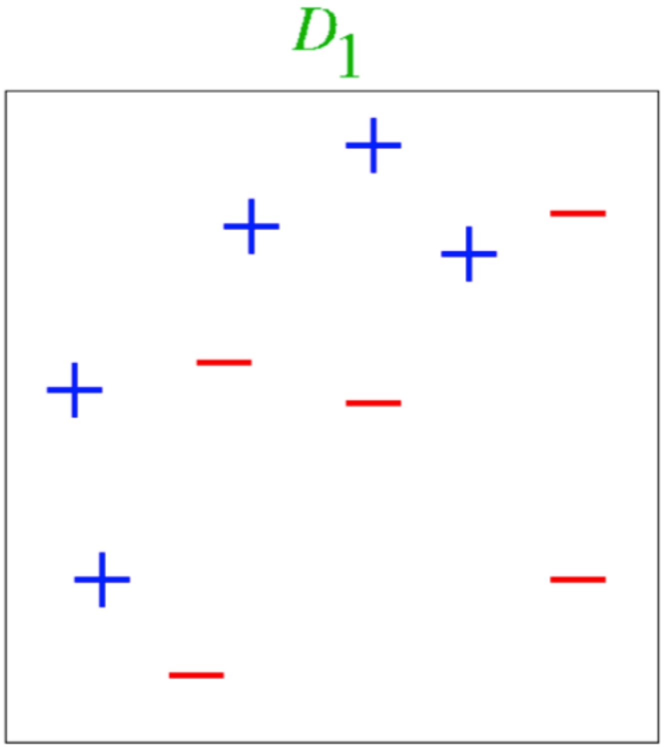
$$w_n^{m+1} = w_n^m \exp\left(-\frac{1}{2} t^{(n)} \alpha_m y_m(\mathbf{x}^{(n)})\right)$$

- Final model

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

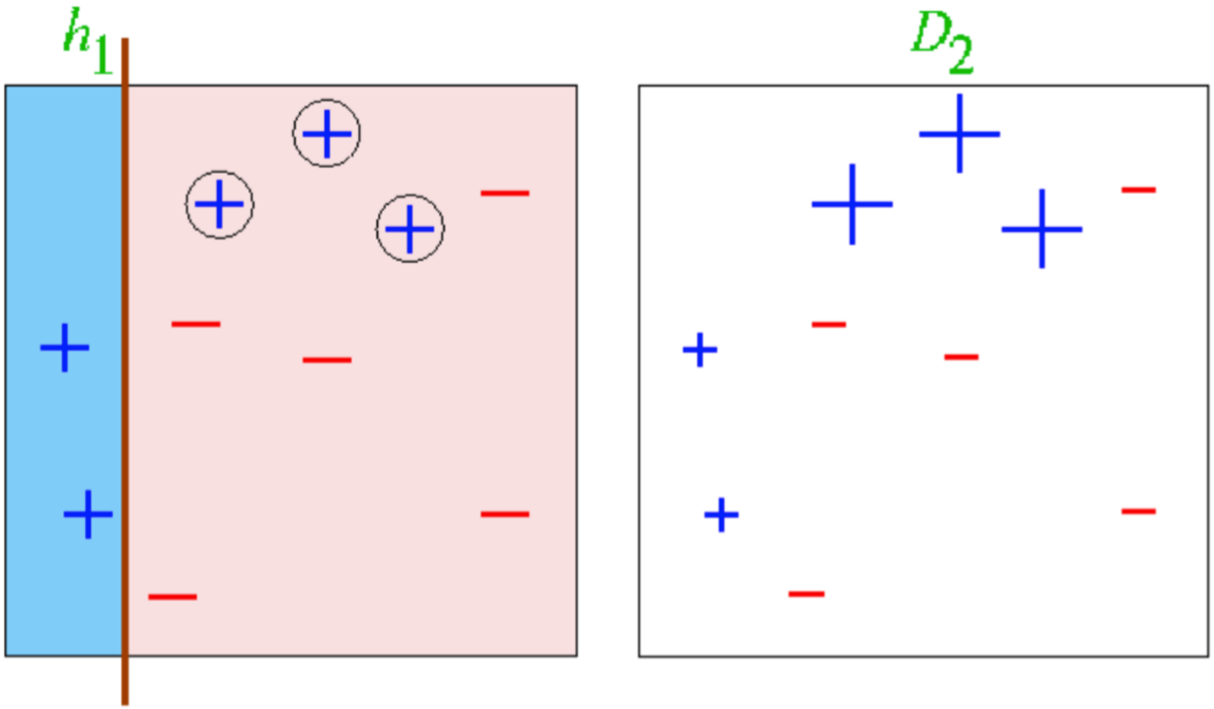
AdaBoost Example

- Training data



AdaBoost Example

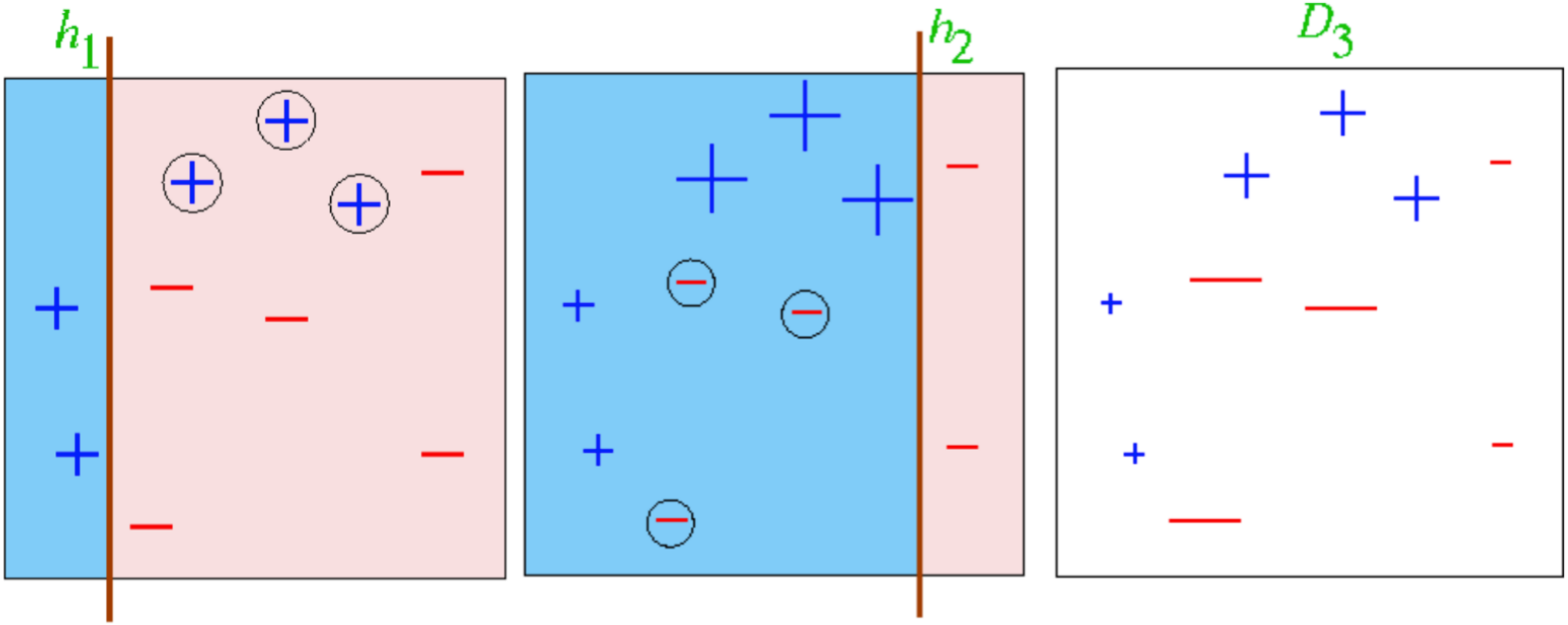
- Round 1



$\varepsilon_1 = 0.300$
$\alpha_1 = 0.847$

AdaBoost Example

- Round 2

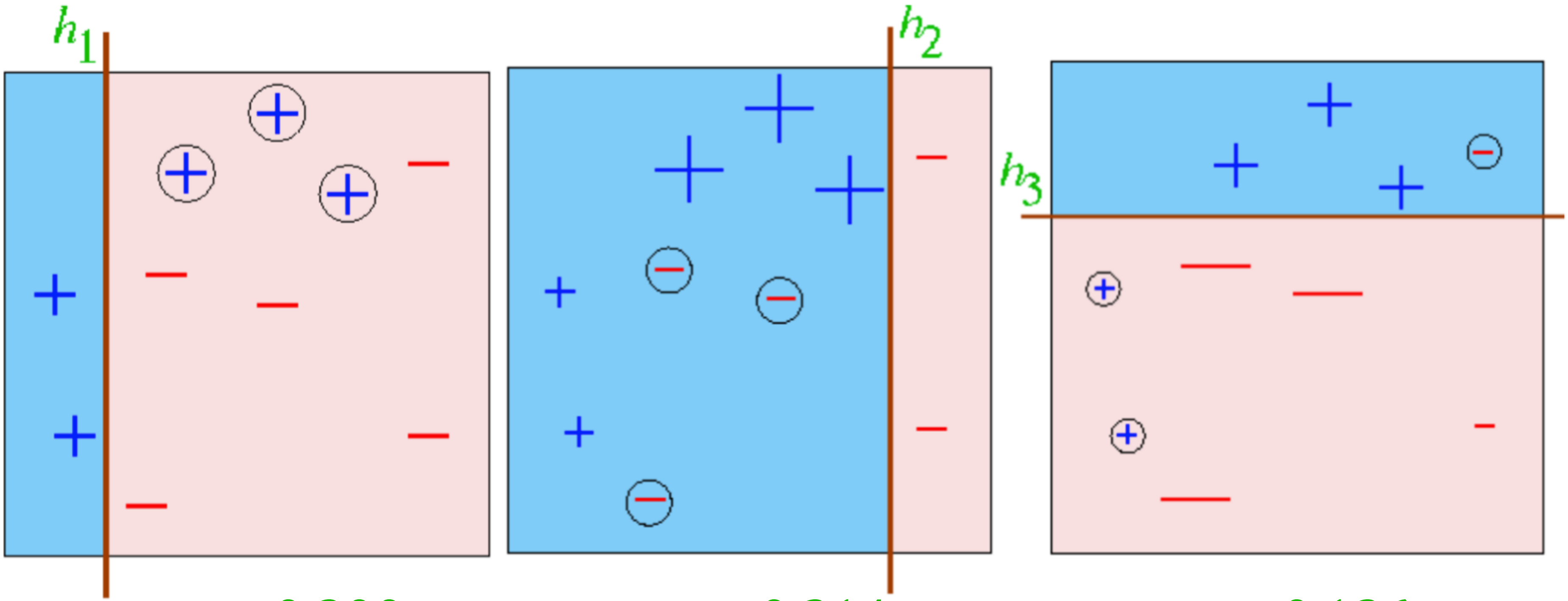


$\varepsilon_1 = 0.300$
$\alpha_1 = 0.847$

$\varepsilon_2 = 0.214$
$\alpha_2 = 1.299$

AdaBoost Example

- Round 3



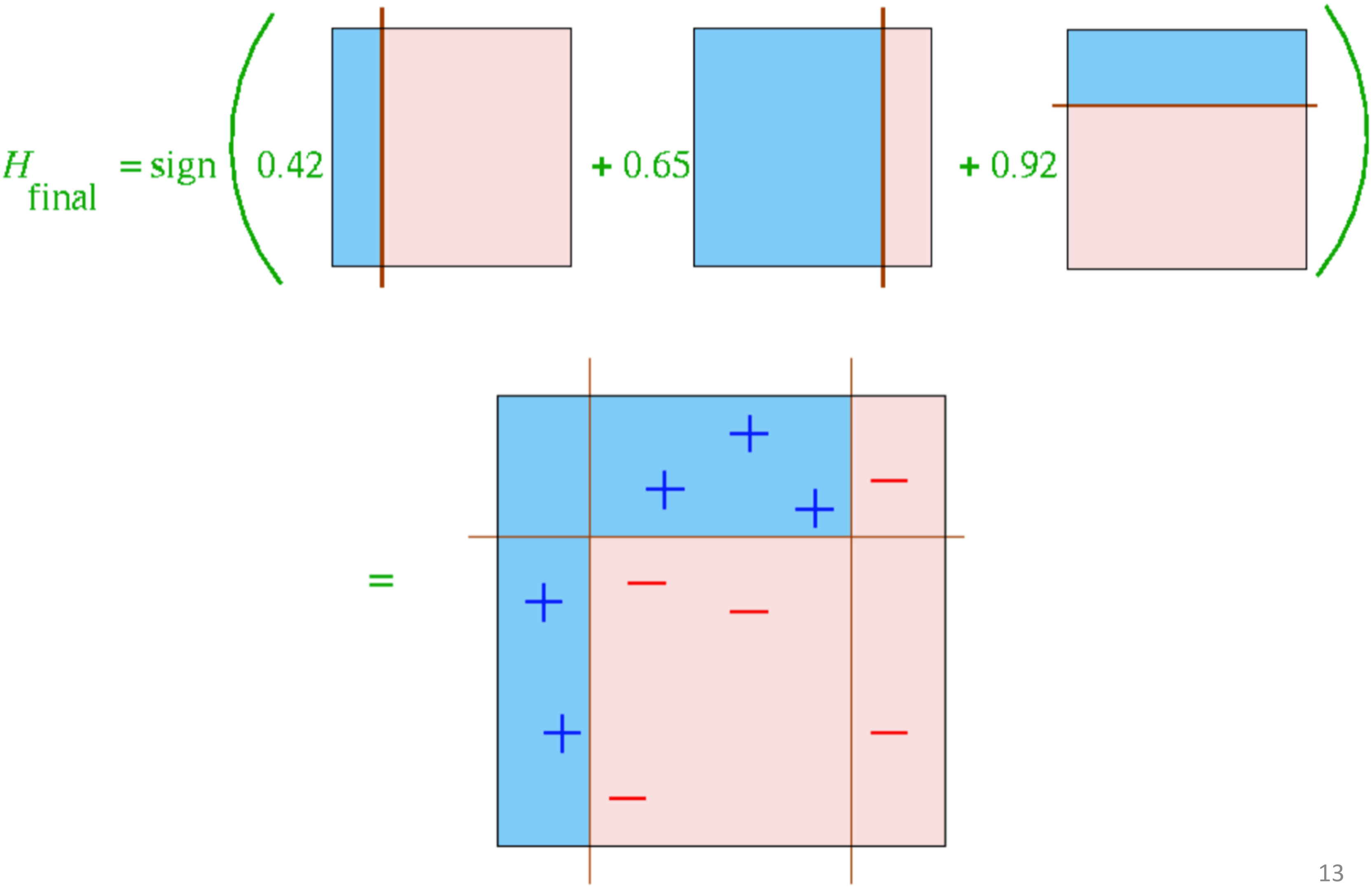
$\varepsilon_1 = 0.300$
$\alpha_1 = 0.847$

$\varepsilon_2 = 0.214$
$\alpha_2 = 1.299$

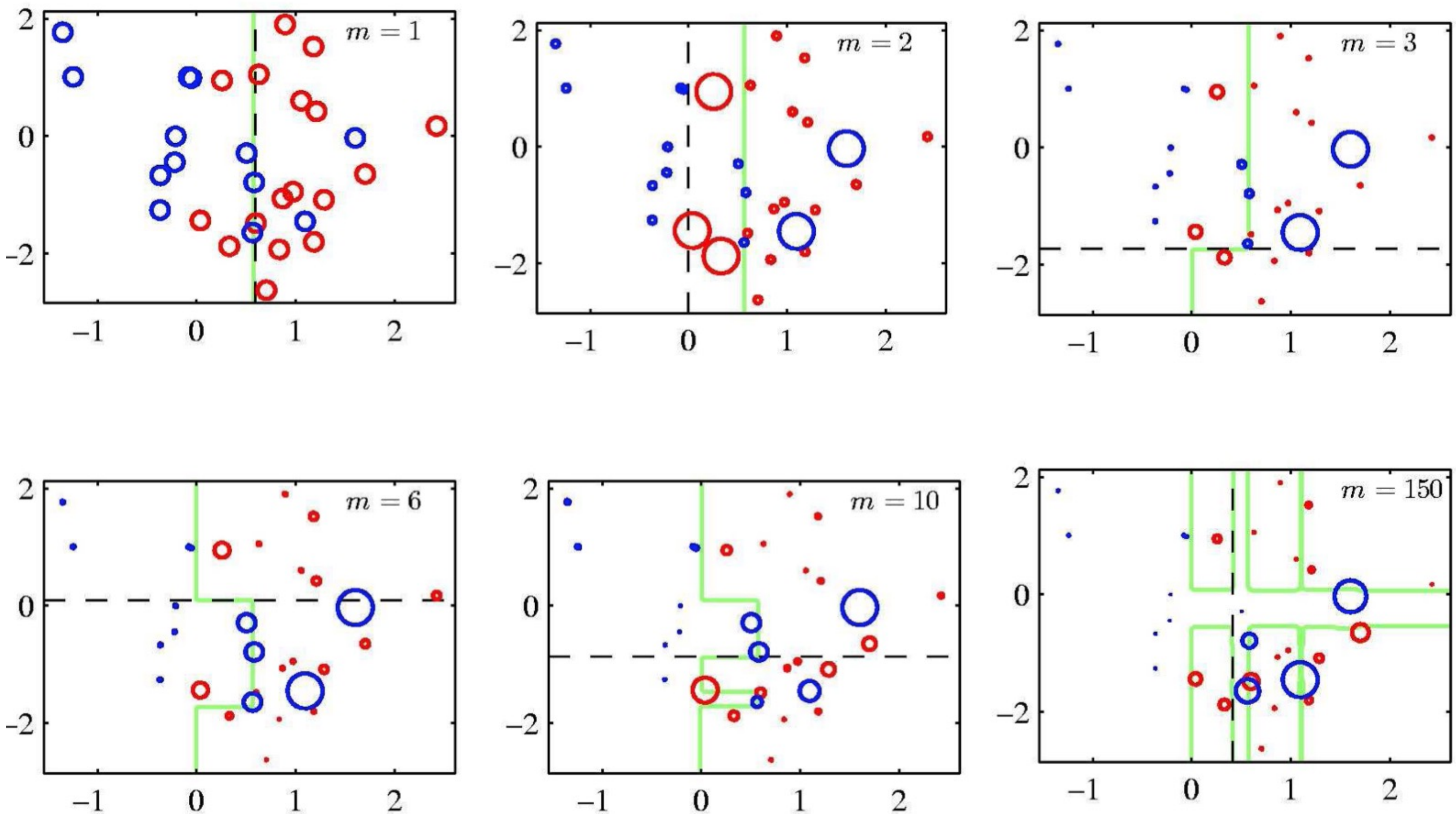
$\varepsilon_3 = 0.136$
$\alpha_3 = 1.846$

AdaBoost Example

- Final classifier



AdaBoost Example



- Each figure shows the number *m* of base learners trained so far, the decision of the most recent learner (dashed black), and the boundary of the ensemble (green)

Questions?

UNIVERSITY OF NORTH CAROLINA
CHARLOTTE