Bias Variance Noise

ECE 4200

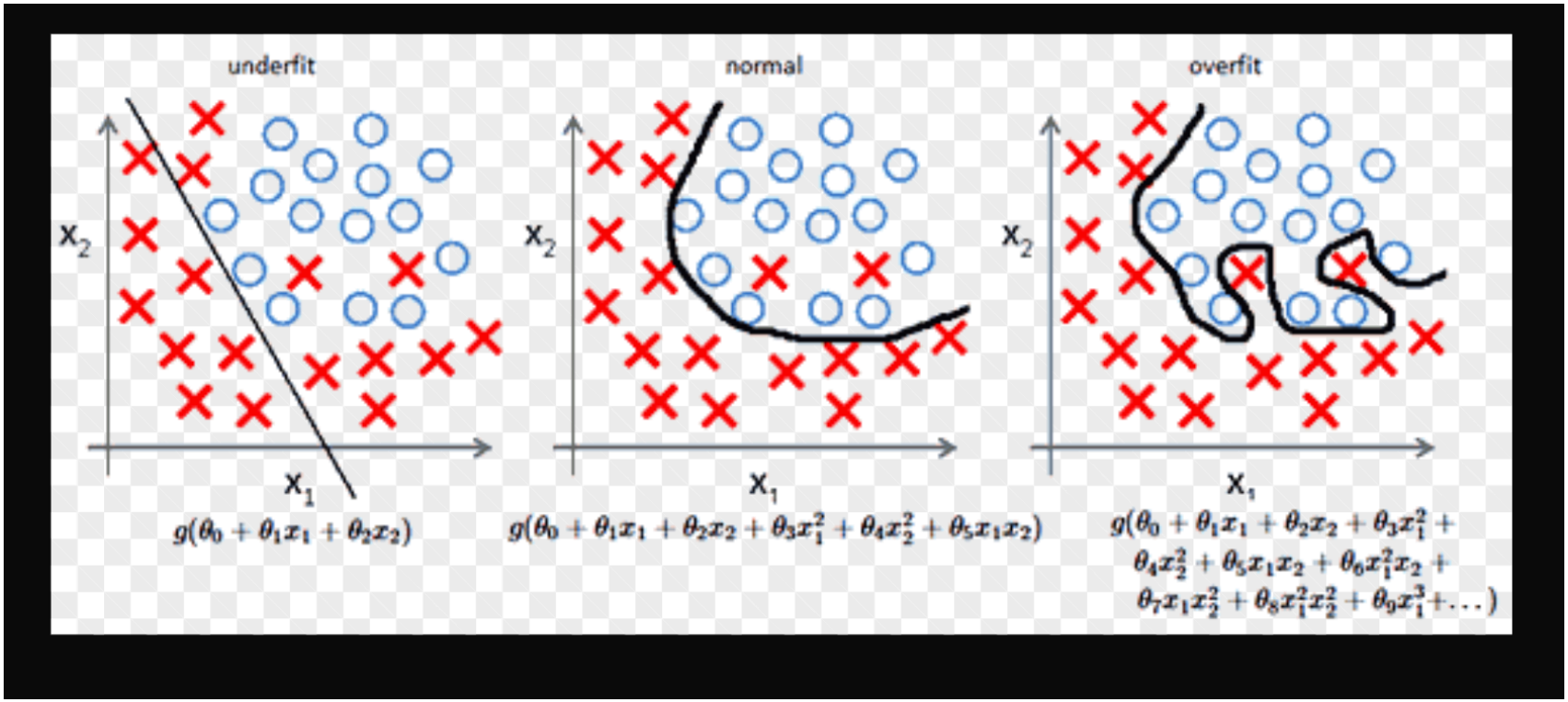
3/4/2020.

Today:-

1. 3. Bias Variance Noise decomposition of error
2. 1. Bootstrap aggregating (Bagging)
3. 2. Boosting.

Overfitting

Model tries to fit to the examples rather than the underlying truth.



Bias variance

The goal of learning: low test error/generalization error

How to understand the test error?

BIAS-VARIANCE TRADE-OFF

Beautiful mathematical decomposition of error

Model

How to study test error: Need to make assumptions on how the data is generated.

Only **ONE ASSUMPTION** :

Each data sample is generated i.i.d. from an **unknown** underlying probabilistic process $\mathrm{P}(\bar{X}, y)$.

$x \times y$

Model

$(\vec{x}, y)$

$\text{error} = (\hat{y} - y)^2$

Only **ONE ASSUMPTION** :

Each data sample is generated i.i.d. from an **unknown** underlying probabilistic process $\mathrm{P}(\vec{x}, y)$.

$$\mathrm{P}(S) = \mathrm{P}(\bar{X}_1, y_1) \ldots \mathrm{P}(\bar{X}_n, y_n).$$

error

For the rest of the day, assume least squares regression, and some class of regression function we want to analyze.

Expected label

Suppose you **know** $\mathrm{P}(\bar{X}, y)$ the entire distribution.

$P(y|\bar{x}) \cdot P(\bar{x})$

Now you are given $\bar{X}$. What y should you predict? **WHY**

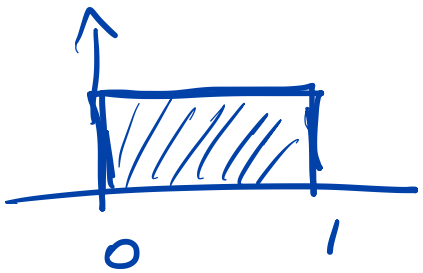
$y \mid \bar{X}$

$$\bar{y}(\bar{X}) = \mathrm{E}[y|\bar{X}] = \int y \, \mathrm{P}(y|\bar{X}) dy$$

$\bar{y}(\bar{X})$ is independent of *S*

If you know $\mathrm{P}(\bar{X}, y)$***, you do not need data!***

z ⟶ $\mathbb{R}$ z



Qn:- Guess $\hat{z}$,

$$\min_{\hat{z}} \; \mathbb{E}_{z}\left[(z-\hat{z})^2\right] \longrightarrow$$

$\mathbb{E}[z]$

$\hat{z}$ ⟶ $\mathbb{E}\left[(z-\hat{z})^2\right] =$ $\mathbb{E}\left[(z-\mathbb{E}[z])^2\right]$ = Var(z)

$+\,\mathbb{E}\left[(\hat{z}-\mathbb{E}[z])^2\right]$

Bias

Set Up

$\mathcal{A}$: algorithm we want to analyze

Linear regression with no regularizer

Linear regression with regularization

SVM's

For the rest of the day, assume regression, and some class of regression function we want to analyze.

Hypothesis

$\mathcal{A}$ takes as input S, and outputs a **hypothesis** h_S.

$$h_S: R^d \rightarrow R$$

Fix the dataset S and $\mathcal{A}$, determines h_S.

Generalization error of h_S

Generalization error:

$$err(h_S) = E_{\bar{X},y\sim P}[(h_S(\bar{X}) - y)^2]$$

Fix a hypothesis, then compute the error on a **new point.**

$$err(h_S) = \int_{\bar{X}} \int_{y} (h_S(\bar{X}) - y)^2 \cdot P\big((\bar{X}, y)\big) dy \, d\, \bar{X},$$

Expected hypothesis

$\mathcal{A}$ takes as input S, and outputs a **hypothesis** h_S.

S is generated randomly from P.

$$\text{Expected hypothesis: } \bar{h}$$

$$\bar{h} = \mathrm{E}_S[\mathcal{A}(S)] = \mathrm{E}_S[h_S]$$

$$\bar{h}(\bar{X}) = \int_S h_S(\bar{X})\, P(S)\, dS$$

Expected value of $h_S(\bar{X})$ over S

Generalization error

Expected generalization error of $\mathcal{A}$:

$$\mathrm{E}_S[\mathcal{A}] = \mathrm{E}_S[err(h_S)]$$

$$\mathrm{E}_S[\mathcal{A}] = \mathrm{E}_S[err(h_S)] = \mathrm{E}_S\left[\mathrm{E}_{(\bar{X},y)\sim P}[(h_S(\bar{X}) - y)^2]\right]$$
$$= \mathrm{E}_{S,(\bar{X},y)\sim P}[(h_S(\bar{X}) - y)^2].$$

Generalization error of h_S

$$\mathrm{E}_S[\mathcal{A}] = \mathrm{E}_{S,\,(\bar{X},y)\sim P}\left[\left(h_S(\bar{X}) - \bar{h}(\bar{X})\right)^2\right]$$
$$+\mathrm{E}_{(\bar{X},y)\sim P}\left[\left(\bar{h}(\bar{X}) - \bar{y}(\bar{X})\right)^2\right]$$
$$+\mathrm{E}_{(\bar{X},y)\sim P}\left[(\bar{y}(\bar{X}) - y)^2\right]$$

Error = Variance + Bias^2 + Noise

Tradeoff – what is the trade-off?

Ensemble methods

Models with high variance, small bias:

Bagging (eg, Decision Trees of Full Depth)

Models with high bias, low variance:

Boosting (Decision stumps, perceptron)

Ensemble Methods

ENSEMBLE METHODS

Ensemble methods

Models with high variance, small bias:

Bagging (eg, Decision Trees of Full Depth)

Models with high bias, low variance:

Boosting (Decision stumps, perceptron)

Ensemble Methods

Wisdom of the crowd

- How to be a millionaire?

  Ask a question to the crowd. Vs

  Ask question to the friend

More likely than not, you get the right answer

Even if each person can be just better than chance, combining all of them can be much more powerful.

Majority

Exercise: Suppose there are 100 people. Each person **independently** votes for **A** with probability 0.6 and for **B** otherwise. (use wolfram alpha)

What is the probability that **B** wins?

$\approx$ 2 %

Ensemble methods

Obtain highly improved classifiers/regressors from weak ones.

Simple ensembles

$$S = \{(\bar{X}_1, y_1), (\bar{X}_n, y_n), \ldots, (\bar{X}_n, y_n)\}, y_i \in \{-1,1\}$$

$y_i \in \mathbb{R}$.

Given new example $(\bar{X}, ?)$.

Train m different predictors classifiers on S.

$\hat{y}_j$: Output of the j th classifier on $\bar{X}$

How to combine the predictions?

Weighted combinations

w_j: weight of the jth predictor ← $w_1, \ldots, w_m$

Larger w_j: more confidence on $\hat{y}_j$.

Final prediction:

$$\hat{y} = \mathbf{sign}(w_1 \hat{y}_1 + w_2 \hat{y}_2 + \ldots + w_m \hat{y}_m) \text{ - cl}$$

$$\hat{y} = \frac{(w_1 \hat{y}_1 + w_2 \hat{y}_2 + \ldots + w_m \hat{y}_m)}{(w_1 + \cdots + w_m)} \text{ - regression.}$$

Simple combinations

$w_j = \frac{1}{m}$: same weight for all predictors

Final prediction:

$\boldsymbol{\hat{y} = \mathbf{sign}(\widehat{y_1} + \widehat{y_2} + \ \ldots + \widehat{y_m})}$ for classification

$\boldsymbol{\hat{y} = \frac{1}{m}(\widehat{y_1} + \widehat{y_2} + \ \ldots + \widehat{y_m})}$ for regression

Success Story

NETFLIX PRIZE

Winning algorithm is a collection of MANY random forests,

BellKor won the first prize

Second team, named **The Ensemble**

Link:

http://courses.washington.edu/css490/2012.Winter/lecture_slides/08a_Netflix_Prize.pptx

BAGGING

BAGGING

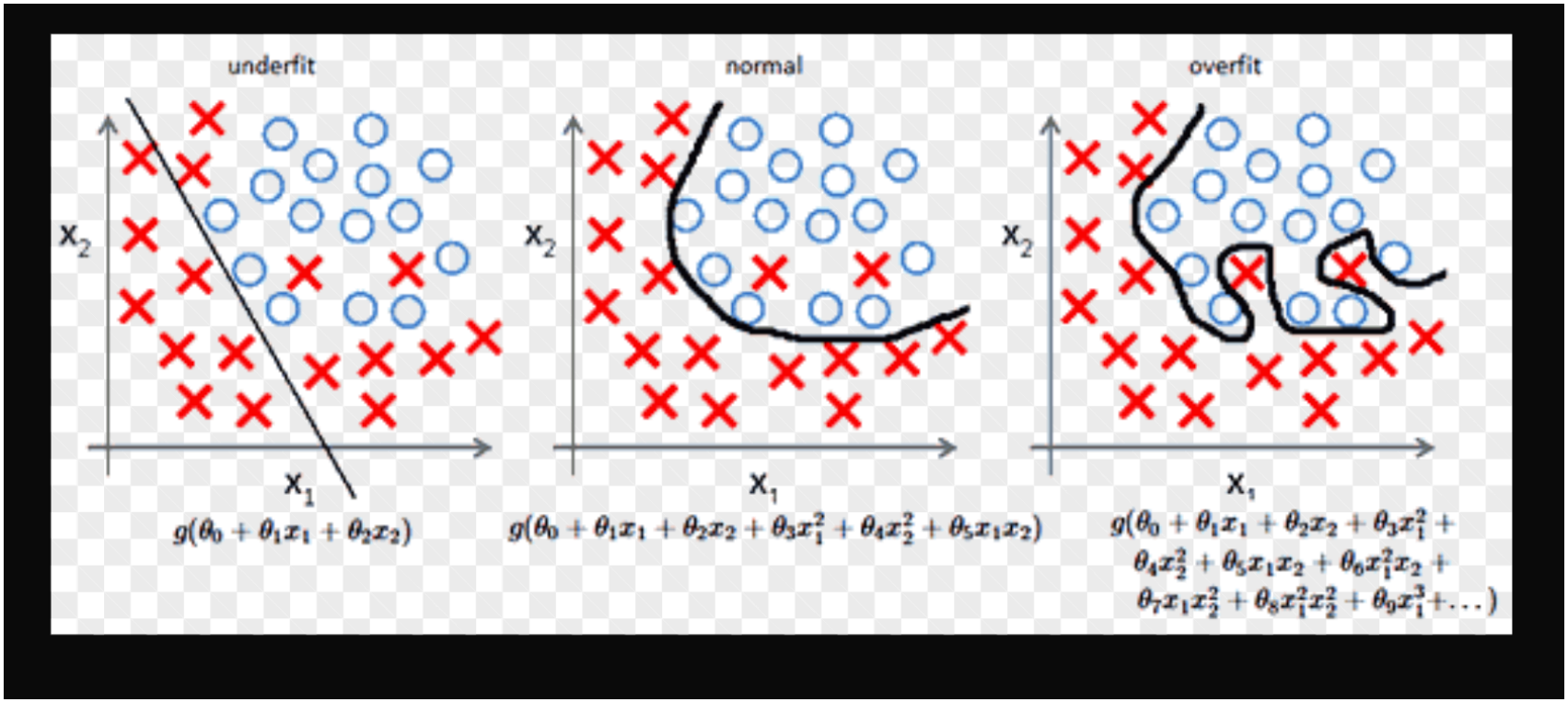
Leo Breiman, “Bagging Predictors”, 1994 (classic).

Stands for “Bootstrap Aggregating”

Beautiful, and simple approach to reduce **overfitting** of models that otherwise tend to overfit

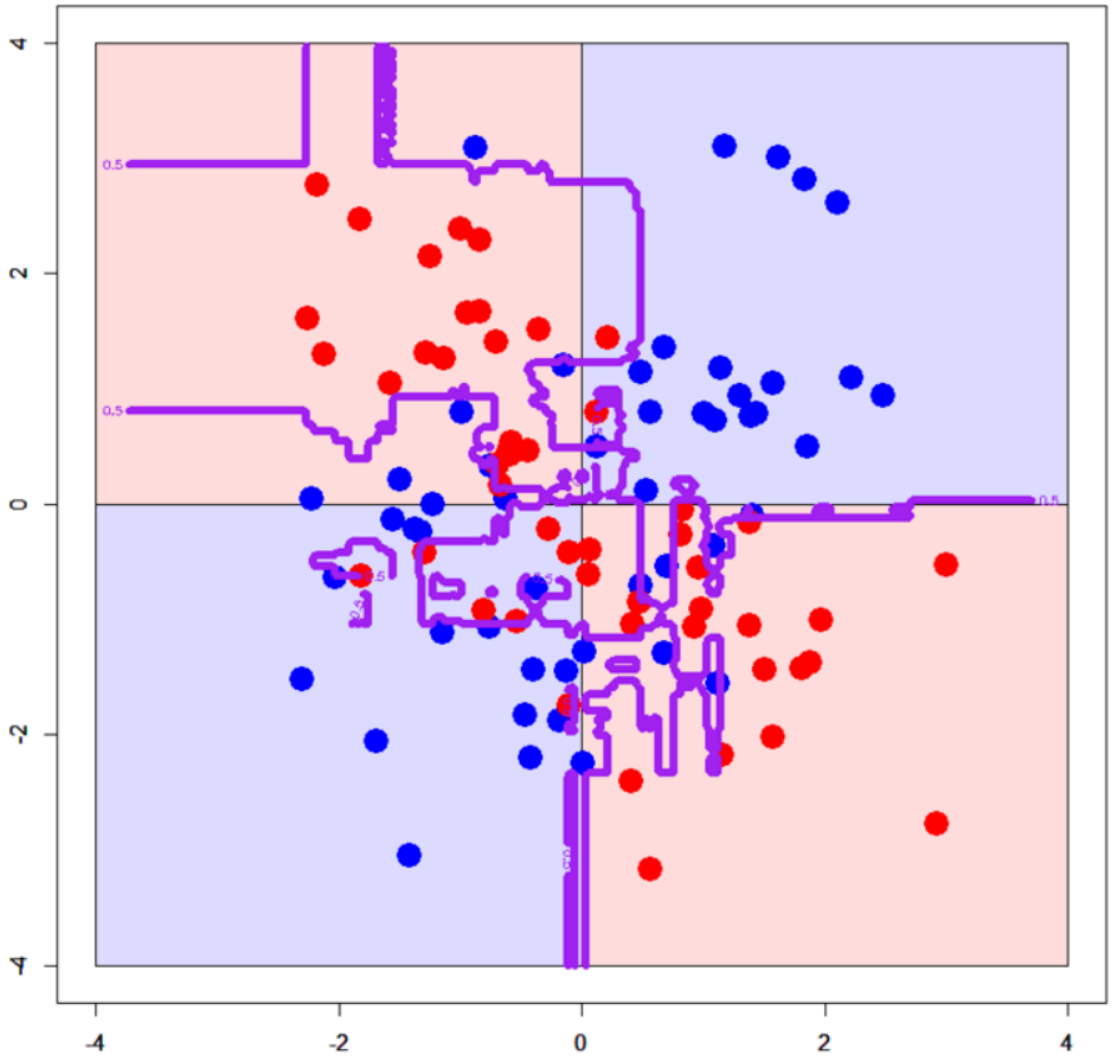
Overfitting

Model tries to fit to the examples rather than the underlying truth.



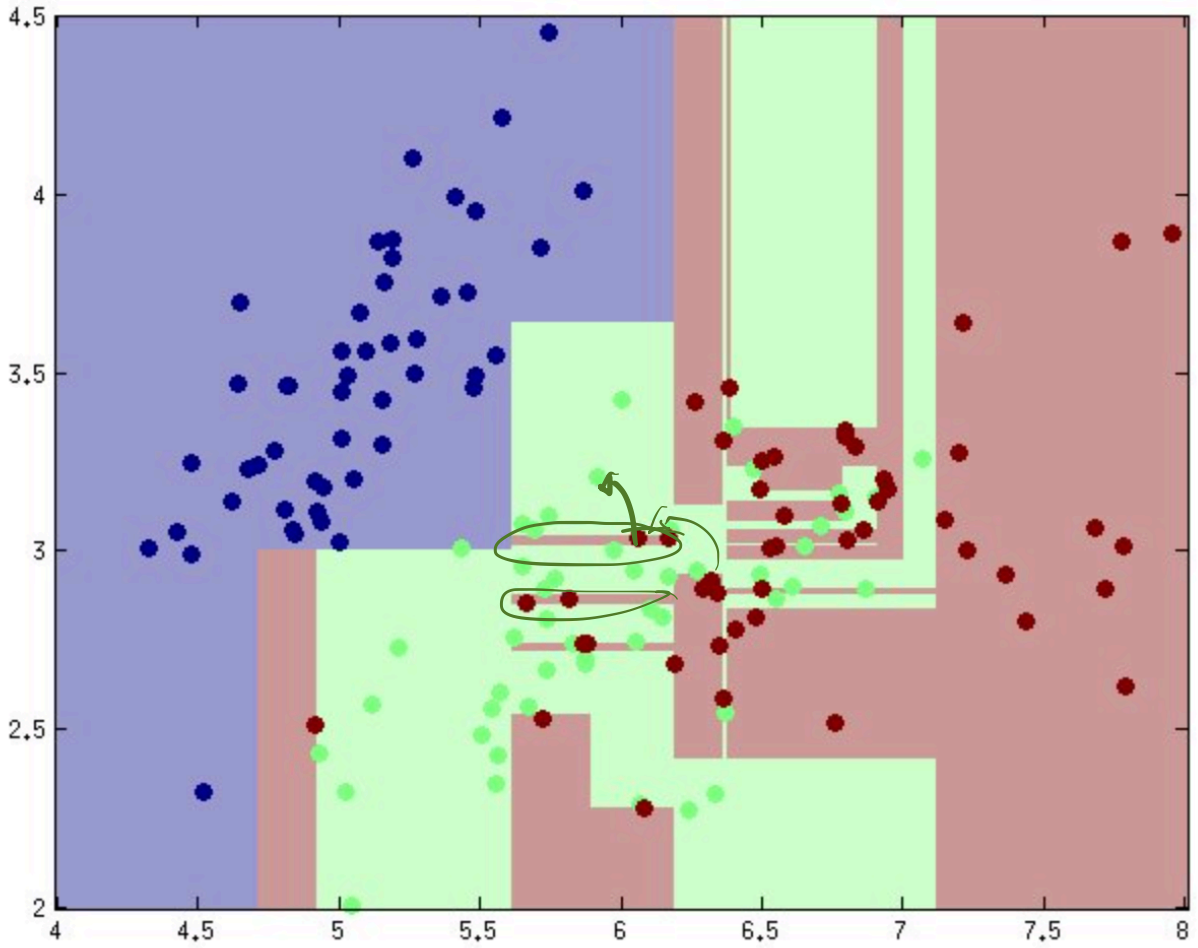
Overfitting decision trees

Model tries to fit to the examples rather than the underlying truth.



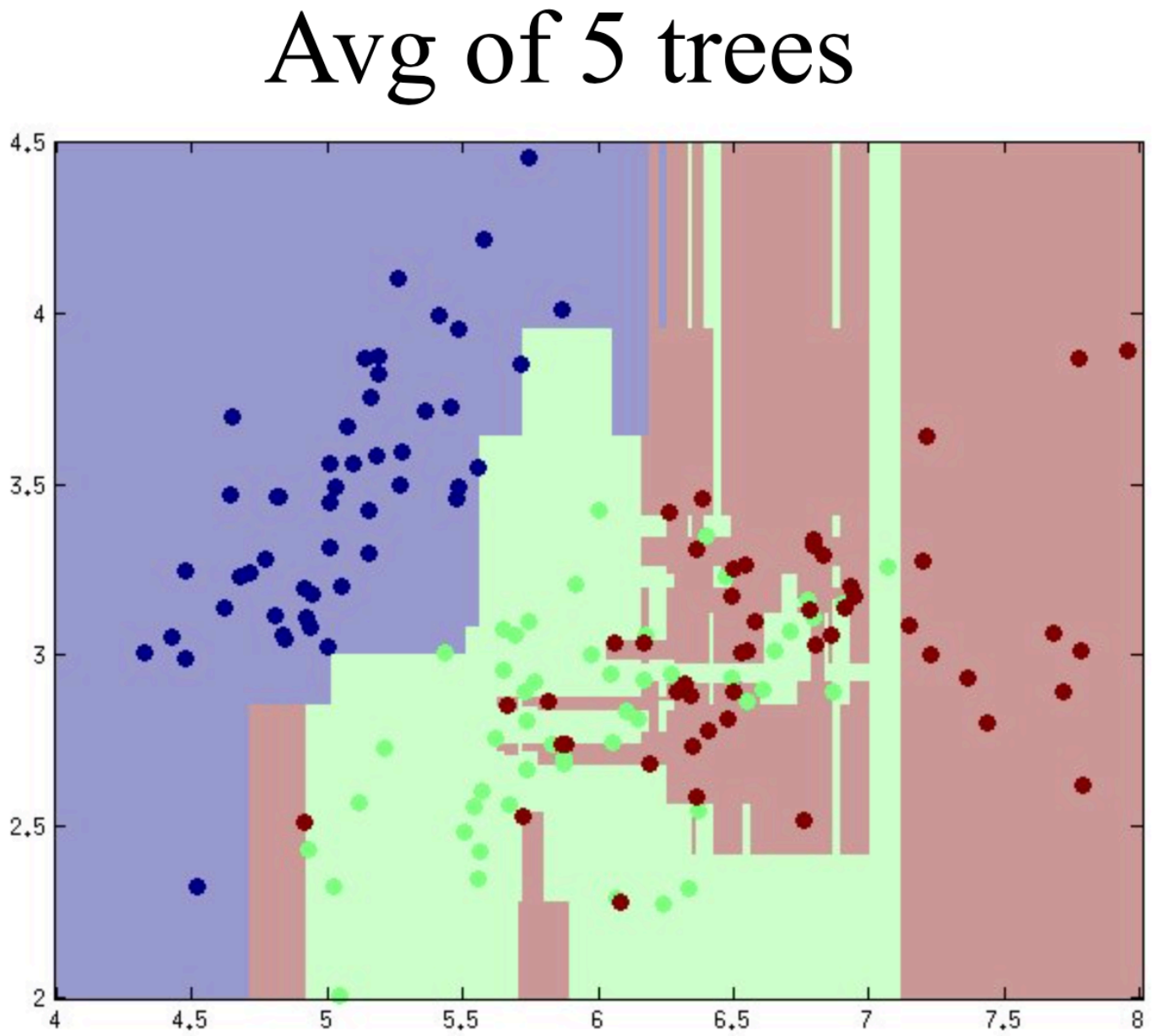
Example on IRIS dataset

IRIS: 150 examples, 3 classes (types of flowers). UCI database, and uci lectures



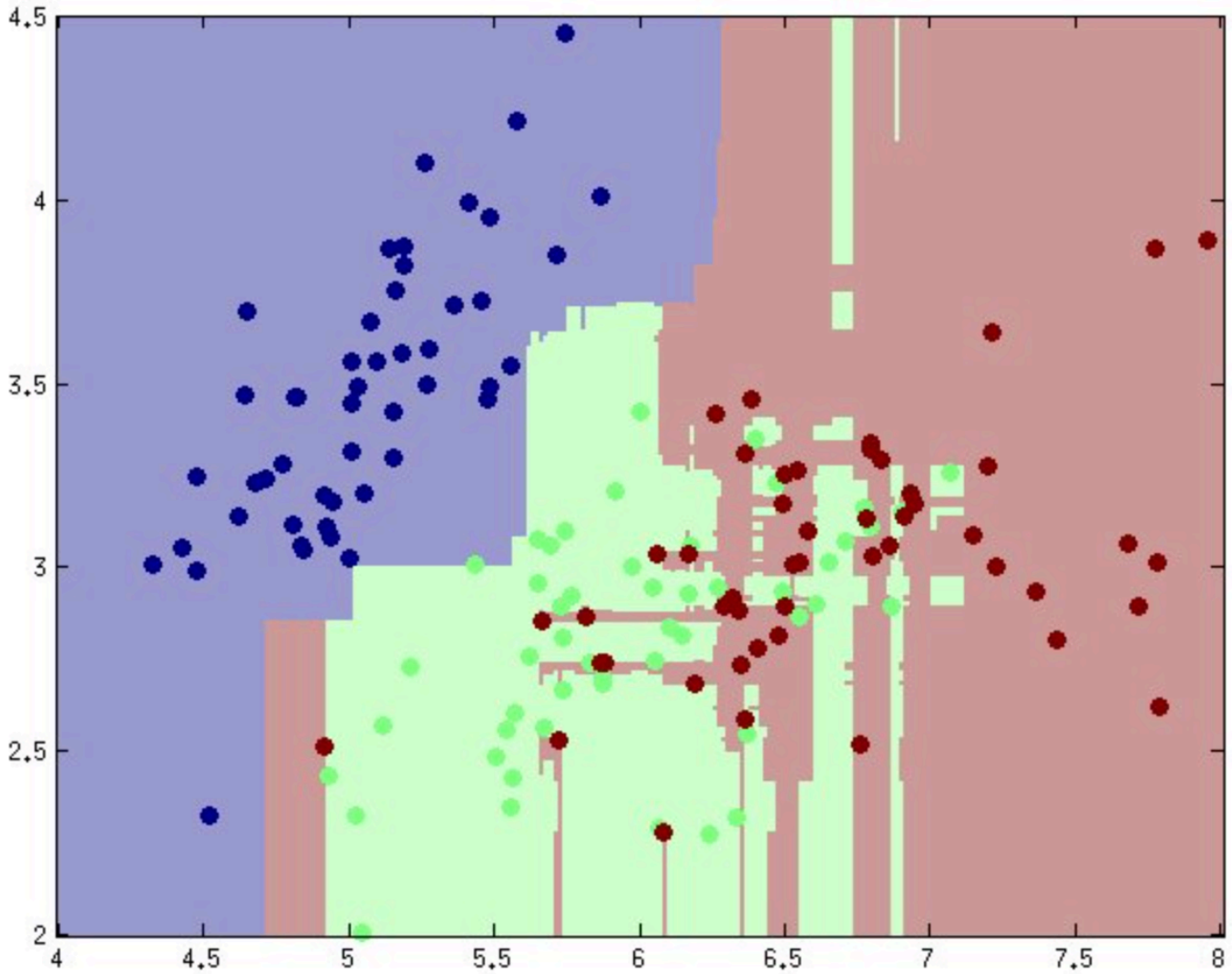
Full data set

Example on IRIS dataset

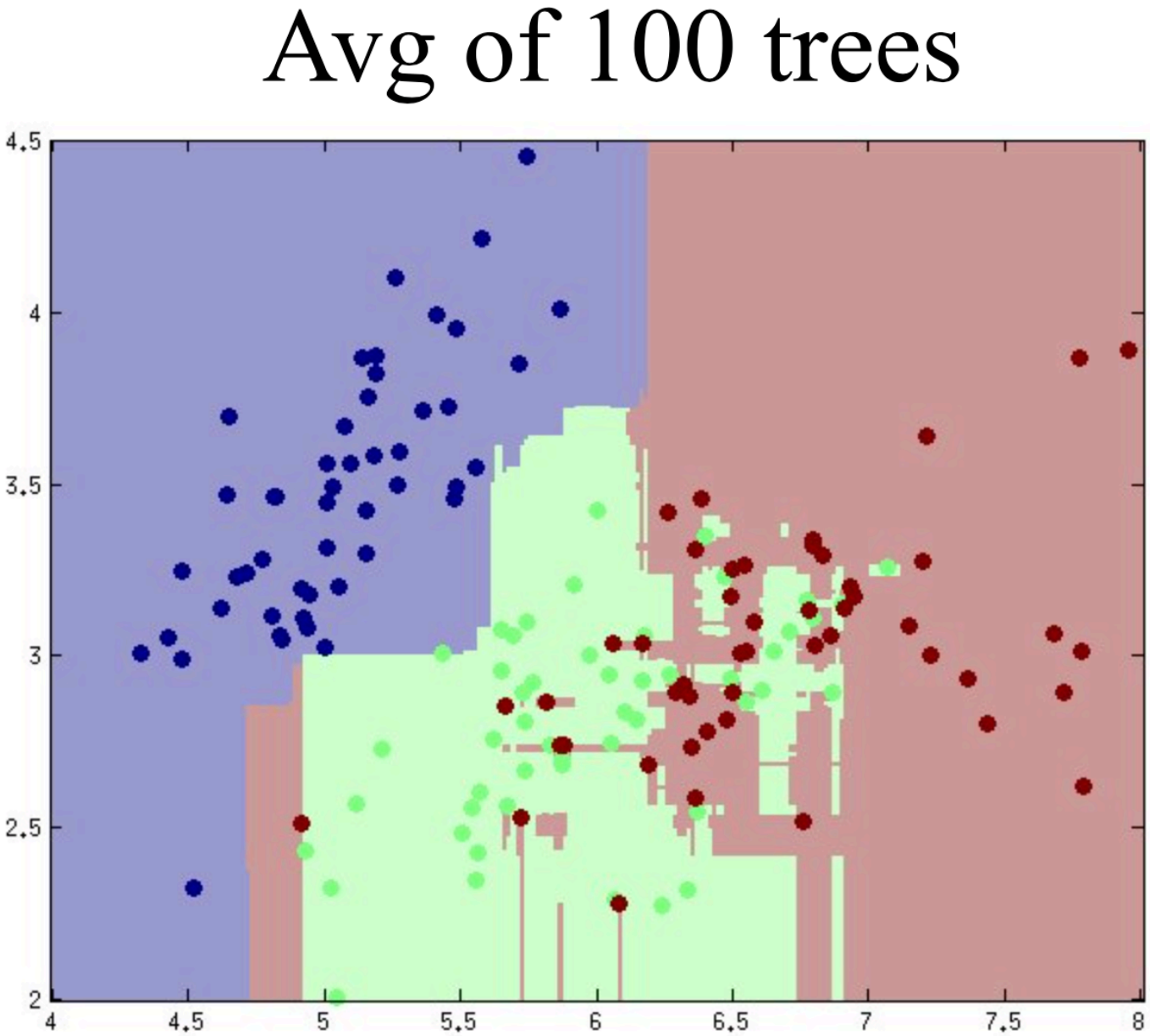


Example on IRIS dataset

Avg of 25 trees



Example on IRIS dataset



Random Forests

With lot of data, DT's tend to be same/similar

Averaging does not help

Further randomness to be added:

- Only allow a subset of features at each level

- Force diverse trees

- Majority vote

Lot of success

CON: computation

BOOSTING

Boosting

Models with high bias:

Boosting:

- **Takes weak classifiers (**Decision stumps, perceptron**)**
- **Combines to output a strong classifier**

Decision stumps are trees with one node!

Simple Rules of Thumb

Weighted error

$h : X \to Y$.

h : a **learner/ hypothesis** (classifier)

p: a distribution over S

Weighted error of h w.r.t. p:

$$err_p(h, S) = \sum_{1 \le i \le n} p(i) \cdot \mathbb{I}\{h(\bar{X}_i) \neq y_i\}$$

Weak learner: takes p, and finds h minimizing weighted error.

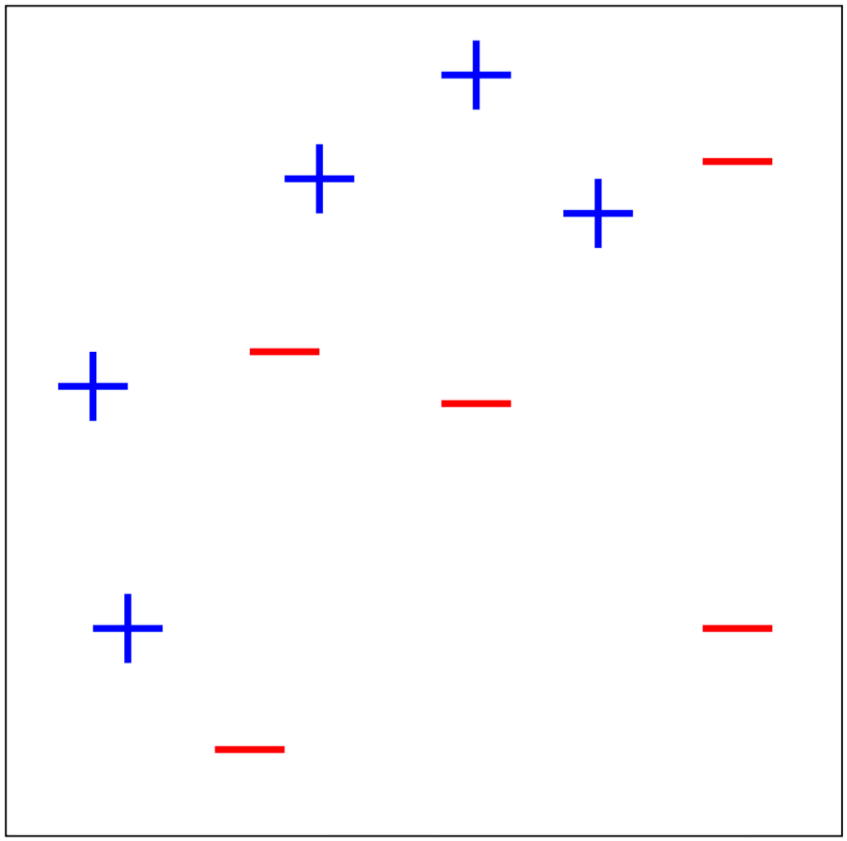
Principle

uniform

- Start with ~~a~~ **distribution** over the training set *S*
- Find weak learner with smallest **weighted** error
- Change the distribution:
  - **Increase** probability of examples **gone wrong**
  - **Reduce** probability of correct examples
- Repeat

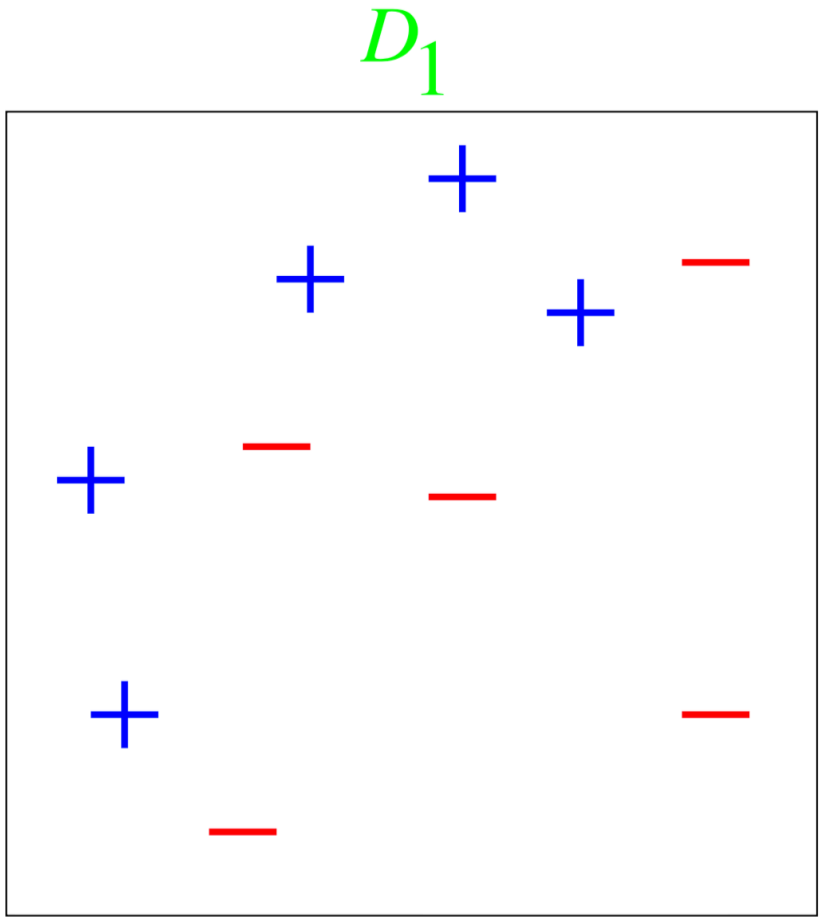
Combine the weak learners in the final stage

Freund Schapire's original example



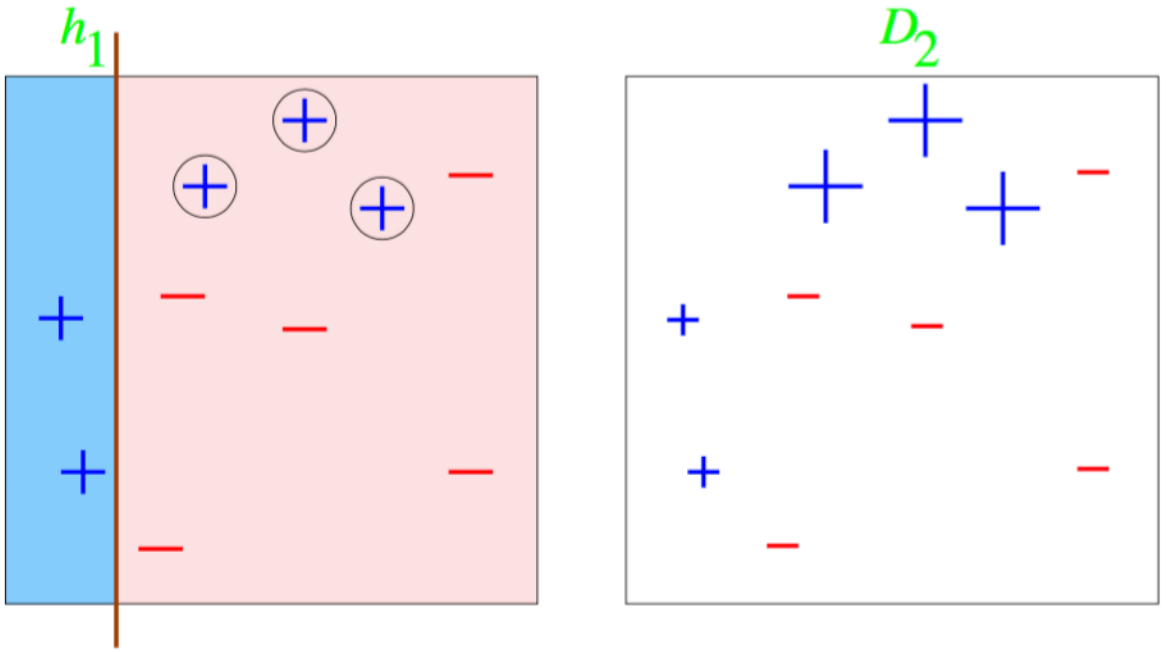
Weak classifiers: vertical/horizontal lines

Freund Schapire's original example



Equal weights

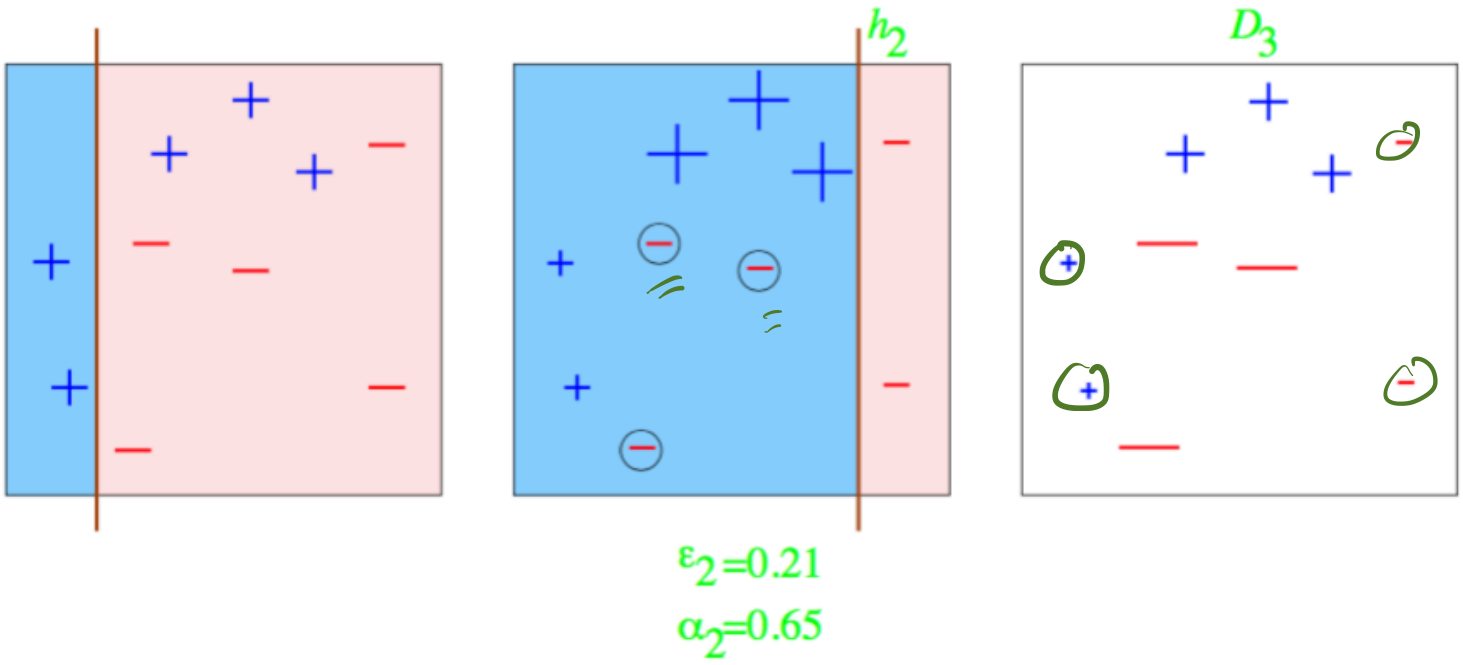
Round 1



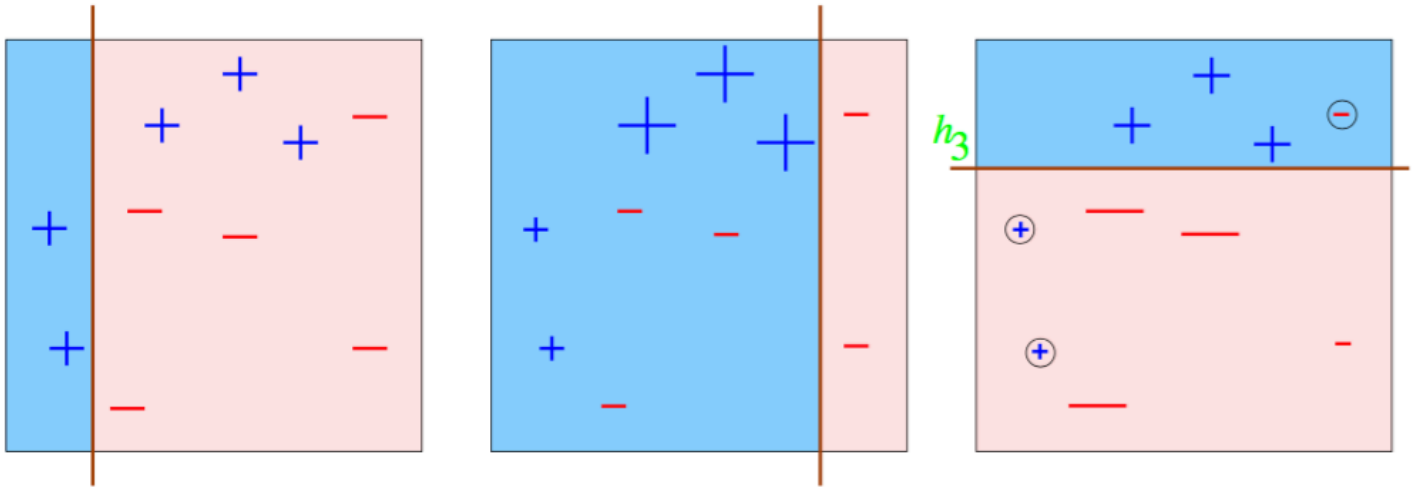
Best line

Increase probability/weights of examples gone wrong

Round 2



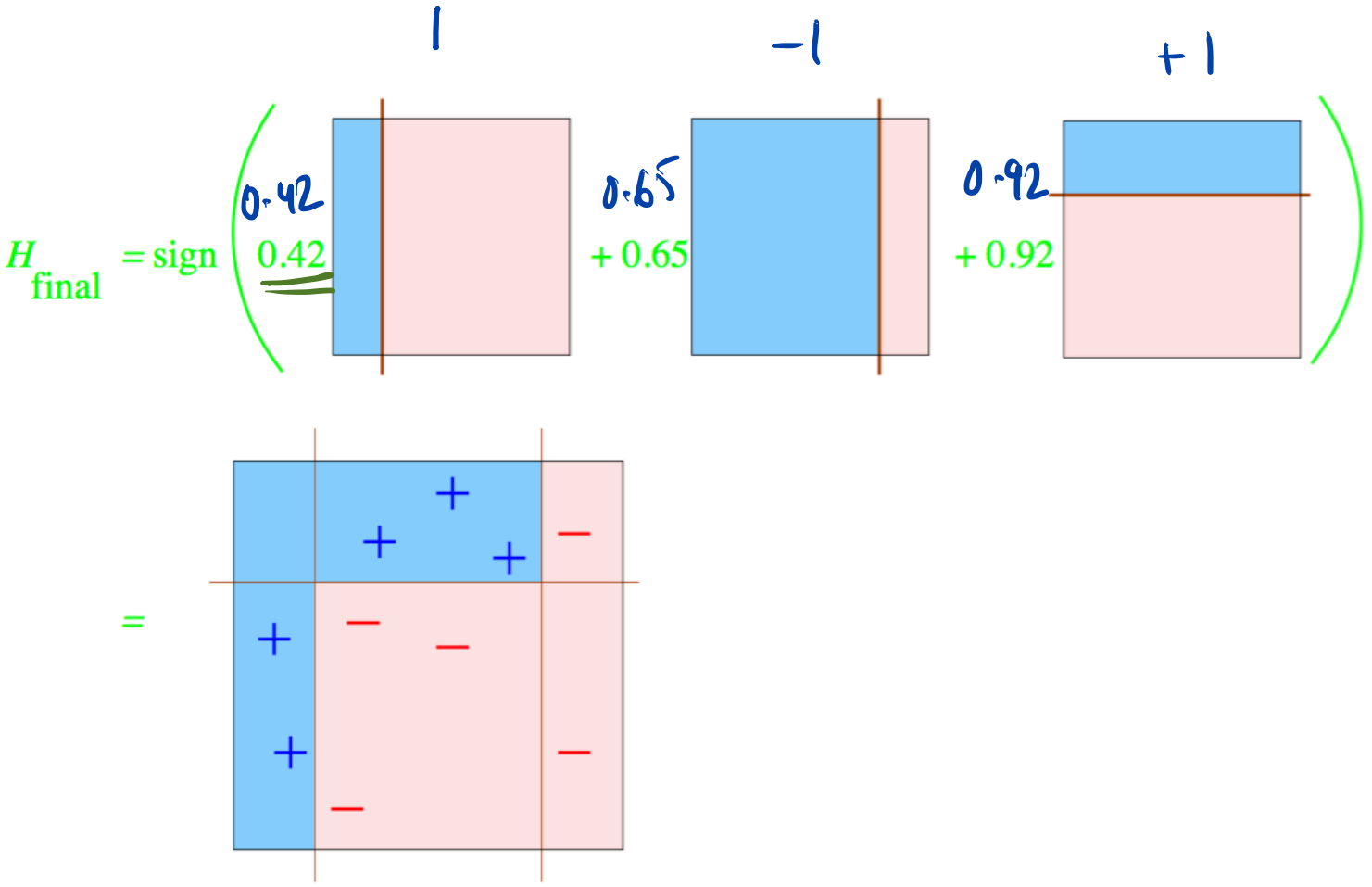
Round 3



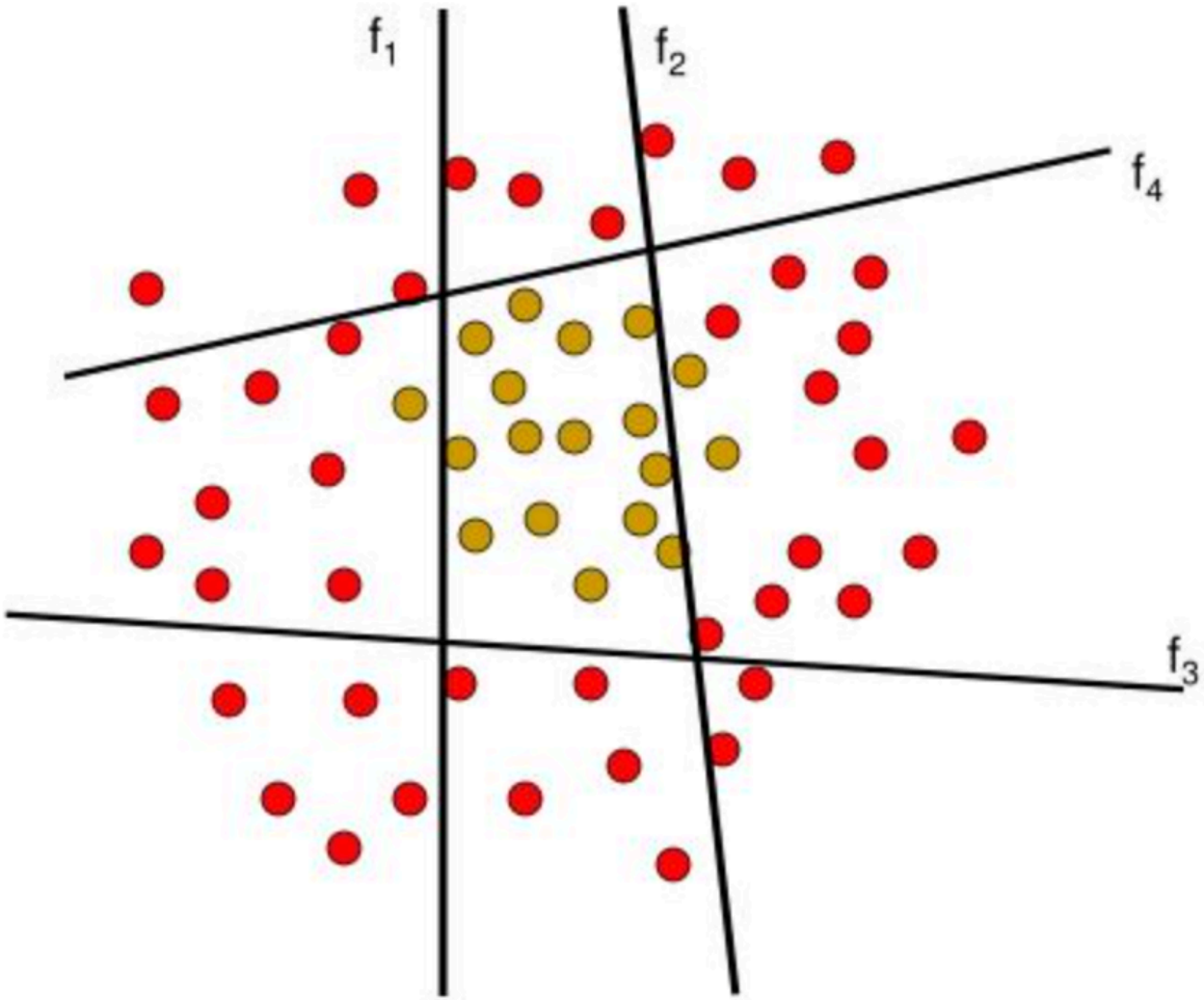
$\varepsilon_3 = 0.14$

$\alpha_3 = 0.92$

Final classifier



With perceptron



AdaBoost algorithm

- Input: $S = \{(\bar{X}_1, y_1), (\bar{X}_n, y_n), \ldots, (\bar{X}_n, y_n)\}$, $y_n \in \{-1, 1\}$
- Initial distribution: $p_1(i) = 1/n$ for each i

For $t = 1, \ldots, T$

- Call **weak learner** w.r.t p_t (h_t)
- **Choose** an $\alpha_t \geq 0$
- Update:

$$p_{t+1}(i) = \begin{cases} p_t(i)\exp(\alpha_t) & \text{if } h_t(\bar{X}_i) \neq y_i \\ p_t(i)\exp(-\alpha_t) & \text{if } h_t(\bar{X}_i) = y_i \end{cases}$$

- Normalize such that $\sum_i p_{t+1}(i) = 1$

Final classifier: sign$\left(\sum_{t=1} \boldsymbol{\alpha_t h_t}(\vec{\boldsymbol{x}})\right)$

How to choose α_t

$h_t \rightarrow$ weak classifier.

Weak learner minimizes:

$$\sum_{1 \le i \le n} p_t(i) \cdot \mathbb{I}\{h(\bar{X}_i) \neq y_i\} = \varepsilon_t$$

ϵ_t: weighted error of the weak learner

$$\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right).$$

$\varepsilon_t \le 1/2$

AdaBoost performance

Suppose $\epsilon_t < \frac{1}{2} - \gamma$ for some gamma.

Then, the training error of final classifier is at most

$$\exp(-2\gamma^2 T)$$

Proof in this discussion session

3/13

Resources for ensemble methods

Trevor Hastie's slides:

http://web.stanford.edu/~hastie/TALKS/boost