CS 360: Machine Learning

Prof. Sara Mathieson

Fall 2020

HAVERFORD
COLLEGE

Admin

- **Lab check in Thursday**: Parts 1&2 complete
- May extend to Friday if that would be helpful!
- Office hours **today: 4:30-6pm**
- Jason TA hours Thurs night

- No class next Tuesday

- Welcome prospective students!

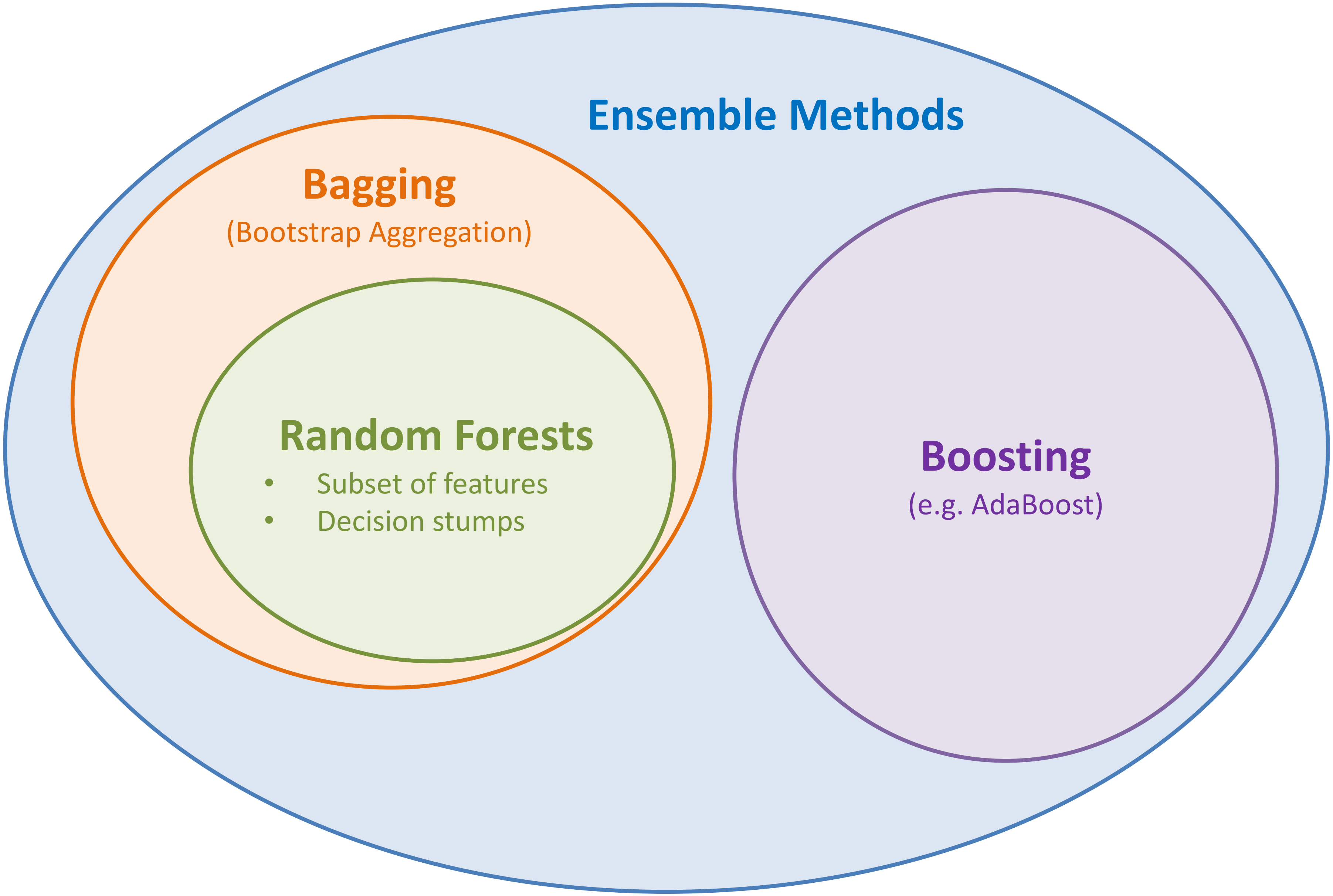
Outline for October 27

- Recap Random Forests
- AdaBoost and weighted entropy
- Perceptron Algorithm

Outline for October 27

- Recap Random Forests
- AdaBoost and weighted entropy
- Perceptron Algorithm

Ensemble Methods big-picture



Random Forests diagram

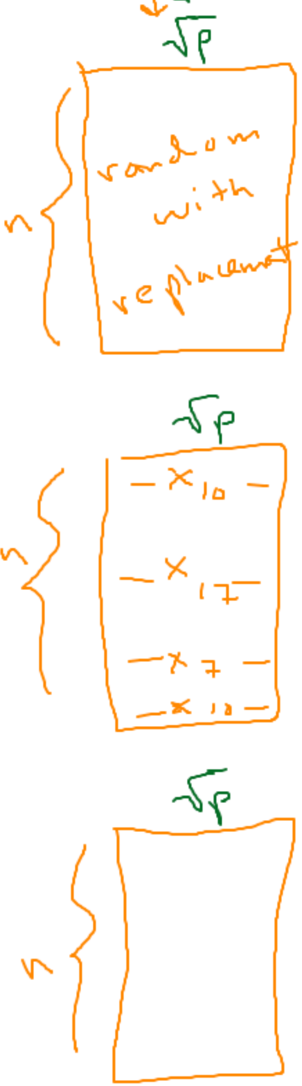
① bootstrapping

② random subset of features

③ classifier = stump (depth = 1)

$T = 3$, $p = 22$

$y \in \{-1, +1\}$

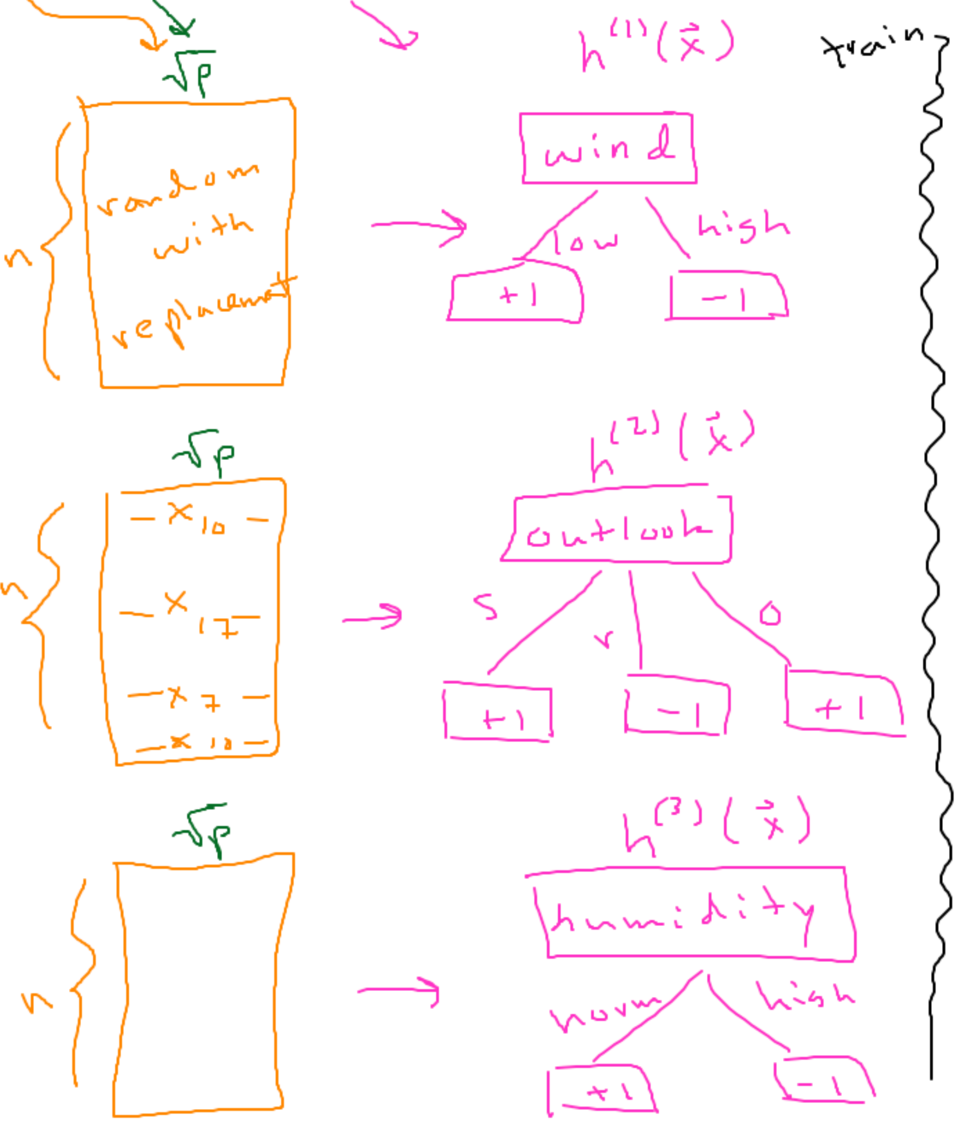


Random Forests diagram

1. bootstrapping
2. random subset of features
3. classifier = stump (depth = 1)

$T = 3$, $p = 22$

$y \in \{-1, +1\}$



Random Forests diagram

1. bootstrapping
2. random subset of features
3. classifier = stump (depth = 1)

$T = 3$, $p = 22$

$y \in \{-1, +1\}$

$\sqrt{p}$

n random with replacement $\rightarrow$ $h^{(1)}(\vec{x})$: wind — low: $+1$; high: -1

$\sqrt{p}$

n x_{10}, x_{17}, x_7, x_{10} $\rightarrow$ $h^{(2)}(\vec{x})$: outlook — s: $+1$; r: -1; o: $+1$

$\sqrt{p}$

n $\rightarrow$ $h^{(3)}(\vec{x})$: humidity — norm: $+1$; high: -1

train | test

out, temp, wind, hum

$\vec{x} = [r, h, low, high]$, $y = +1$

$h^{(1)}(\vec{x}) = +1$

$h^{(2)}(\vec{x}) = -1$

$h^{(3)}(\vec{x}) = -1$

$\boxed{h(\vec{x}) = -1}$

Outline for October 27

- Recap Random Forests
- AdaBoost and weighted entropy

Note: AdaBoost is NOT an extension of Random Forests
However! We are using Decision Stumps for both

- Perceptron Algorithm

AdaBoost (adaptive boosting)

Initialization

- Assign uniform weights to all training data points: $w_i^{(1)} = \frac{1}{n}$ for $i = 1, 2, \cdots, n$. Note that we require the weights to sum to 1.

Adaptive Procedure

no bootstrap

For $t = 1, 2, \cdots T$, use the following procedure to find a new classifier and update the weights on the training examples:

(a) Fit a classifier to the weighted training set. We will call this classifier $h^{(t)}(\boldsymbol{x})$.

(b) Compute weighted classification *error* on the training set:

$$\epsilon_t = \sum_{i=1}^{n} w_i^{(t)} \mathbb{1}\left(y_i \neq h^{(t)}(\boldsymbol{x}_i)\right)$$

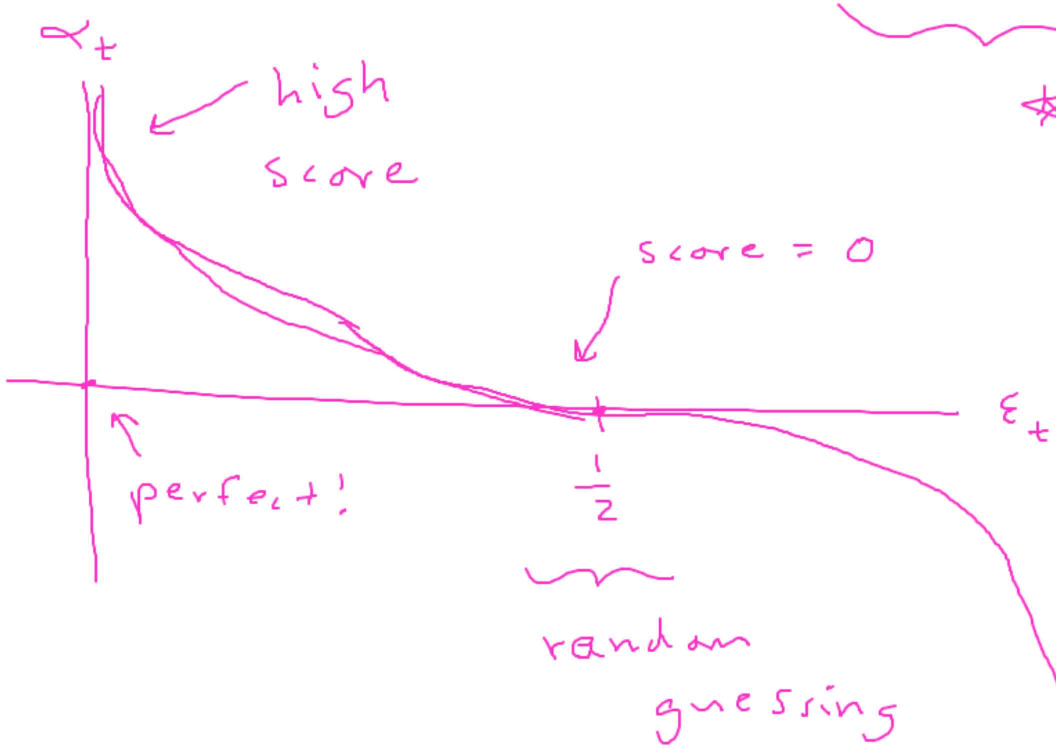
Note that since the weights sum to 1, $0 \leq \epsilon_t \leq 1$. However, since we are in a binary classification scenario, we should never have an error greater than 0.5.

AdaBoost (adaptive boosting)

(c) Compute the *score* of the classifier:

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

The score is 0 when $\epsilon_t = \frac{1}{2}$ (random guessing). As $\epsilon_t \to 0$, $\alpha_t \to \infty$ (i.e. a very good classifier).



AdaBoost – updating weights

$y_i \in \{-1, +1\}$

(d) update weights on examples

$$w_i^{(t+1)} = c_t \cdot w_i^{(t)} \exp\left(-y_i \alpha_t h^{(t)}(\vec{x}_i)\right)$$

normalizer so weight sum to 1

(1) if $y_i = h^{(t)}(\vec{x}_i) \Rightarrow e^{-\alpha_t}$

(2) if $y_i \neq h^{(t)}(\vec{x}_i) \Rightarrow e^{\alpha_t}$

(1) right $\Rightarrow$ decrease

(2) wrong $\Rightarrow$ increase weight!

AdaBoost – updating weights

(d) update weights on examples

$y_i \in \{-1, +1\}$

$$w_i^{\cancel{(t+1)}} = c_t \cdot w_i^{\cancel{(t)}} \exp\left(-y_i \alpha_t h^{(t)}(\vec{x}_i)\right)$$

each example

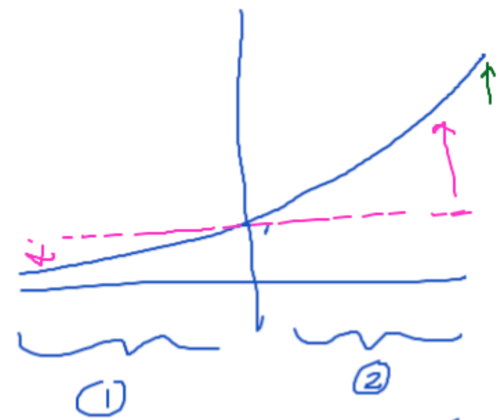
normalizer so weights sum to 1

$$\text{(1) if } \overbrace{y_i = h^{(t)}(\vec{x}_i)}^{\text{right}} \Rightarrow e^{-\alpha_t}$$

$$\text{(2) if } \underbrace{y_i \neq h^{(t)}(\vec{x}_i)}_{\text{wrong}} \Rightarrow e^{\alpha_t}$$

$$c = \frac{1}{w_i^{(t)} \exp(\qquad)}$$

$\alpha_t = 0$?



(1) right ⇒ decrease

(2) wrong ⇒ increase weight!

AdaBoost implementation

for $t = 1 \ldots T$:

$h^{(t)} = \boxed{DS(\ldots)}$

$\varepsilon_t = \boxed{}$ error

$\alpha_t = \boxed{}$ score

↑ wrong

↓ right

for $i = 1 \ldots n$

$wp[i] = w_i \cdot \exp(\quad)$

$c \mathrel{+}=$

norm

for $i = 1 \ldots n$

$w_i = c \cdot wp[i]$

Decision Trees with Weighted Entropy

cond.
Entropy

$$H(Y \mid x_j = v) = -\sum_{\underbrace{c \in \text{vals}(y)}_{\{-1,1\}}} \overbrace{P(Y=c \mid x_j = v)}^{\text{weighted}} \log_2 P(Y=c \mid x_j = v)$$

Bayes rule

$$P(Y=c \mid x_j = v) = \frac{P(Y=c, x_j = v)}{P(x_j = v)}$$

count of examples with $Y = c$ & $x_j = v$

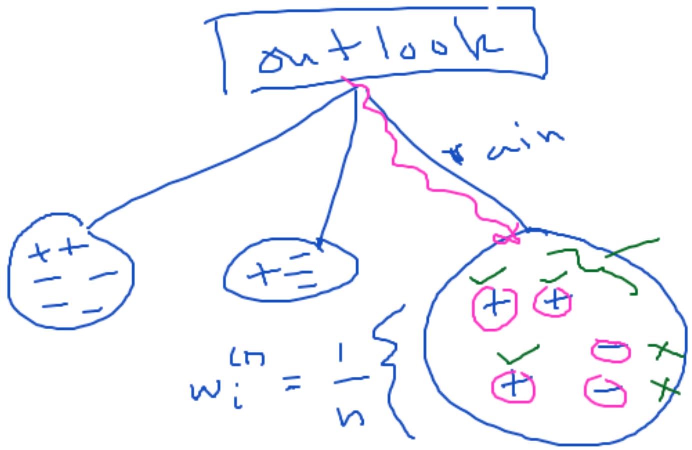
$w_i^{(1)} = \frac{1}{n}$

$$= \frac{\sum_{i=1}^{n} w_i^{(t)} \mathbb{1}(y_i = c, x_{ij} = v)}{\sum_{i=1}^{n} w_i^{(t)} \mathbb{1}(x_{ij} = v)}$$

weight on ith example

Decision Trees with Weighted Entropy: example

thresh = 0.5



$w_i^{(1)} = \frac{1}{n}$

$$p(\text{pos}) = \frac{\sum_{\text{leaf}} w_i^{(t)} \, \mathbb{1}(y_i = 1)}{\sum_{\text{leaf}} w_i^{(t)}} = \frac{\left(\frac{1}{17}\right)\cdot 3}{\left(\frac{1}{17}\right)\cdot 5} = \boxed{\frac{3}{5}}$$

$\boxed{y = 1}$

$n = 17$

~~much~~ later

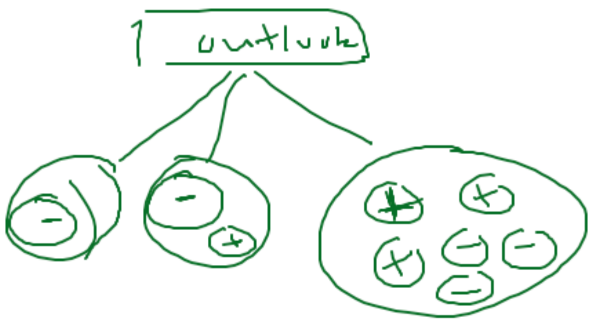
$t = 2$

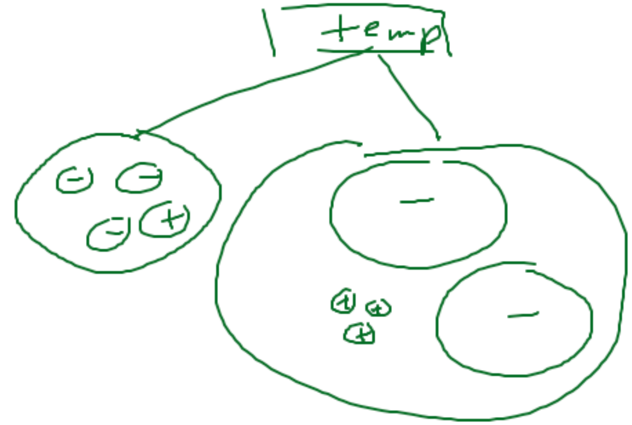
$\frac{1}{12}$

$\frac{1}{4}$

$$\text{pos}(+) = \frac{\frac{1}{12} + \frac{1}{12} + \frac{1}{12}}{\frac{1}{12}\cdot 3 + \frac{1}{\cancel{4}\ 12}\cdot 2\cdot 3} = \frac{3}{3+6} = \boxed{\frac{1}{3}}$$

Decision Trees with Weighted Entropy: example





heterogeneous

$p(+) = \frac{1}{2}$

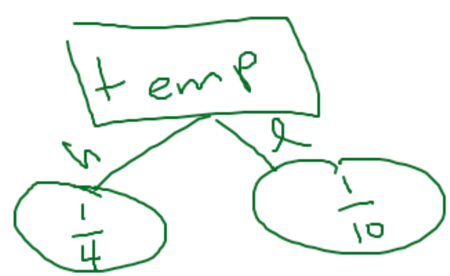
$p(-) = \frac{1}{2}$

homogeneous

$p(+) = \frac{1}{10}$

$p(-) = \frac{9}{10}$

retain



Decision Trees with Weighted Entropy: example

output? $\Rightarrow$ def trainAB():

$[h^{(1)}, h^{(2)} \ldots]$

return (1) list of stumps

$[\alpha_1, \alpha_2, \ldots]$

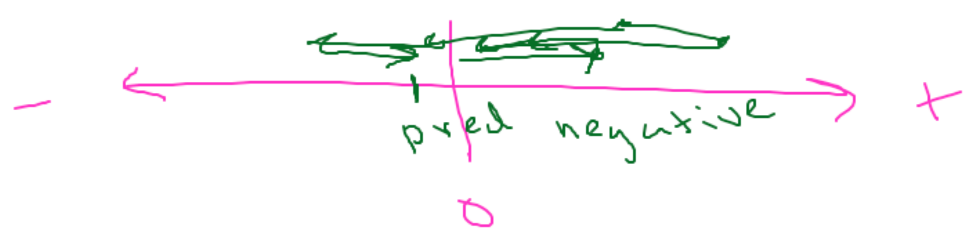
(2) list of scores

train

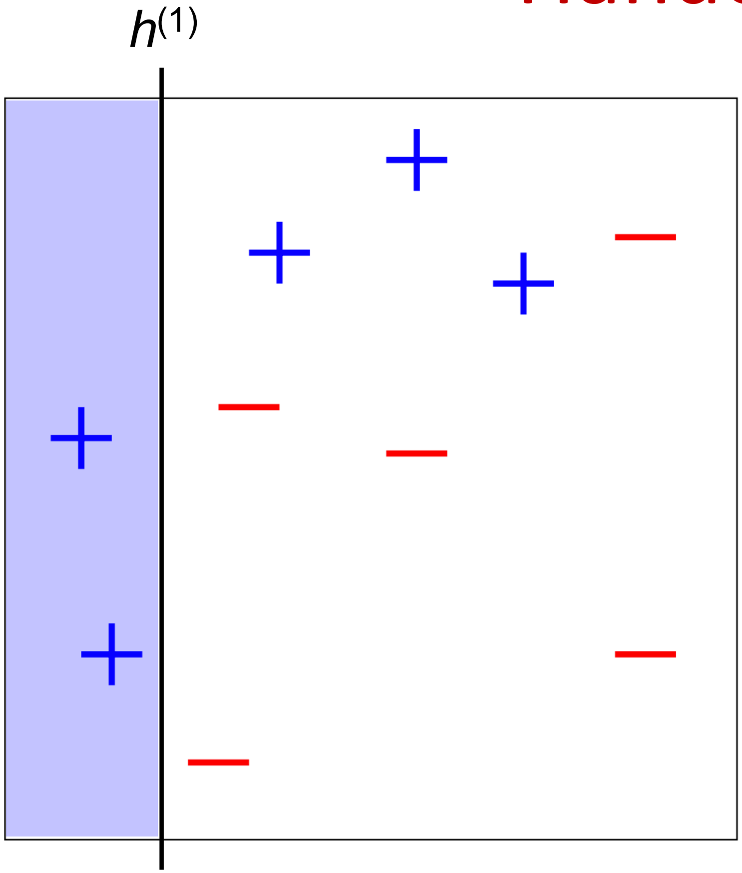
test

$[0, \infty)$

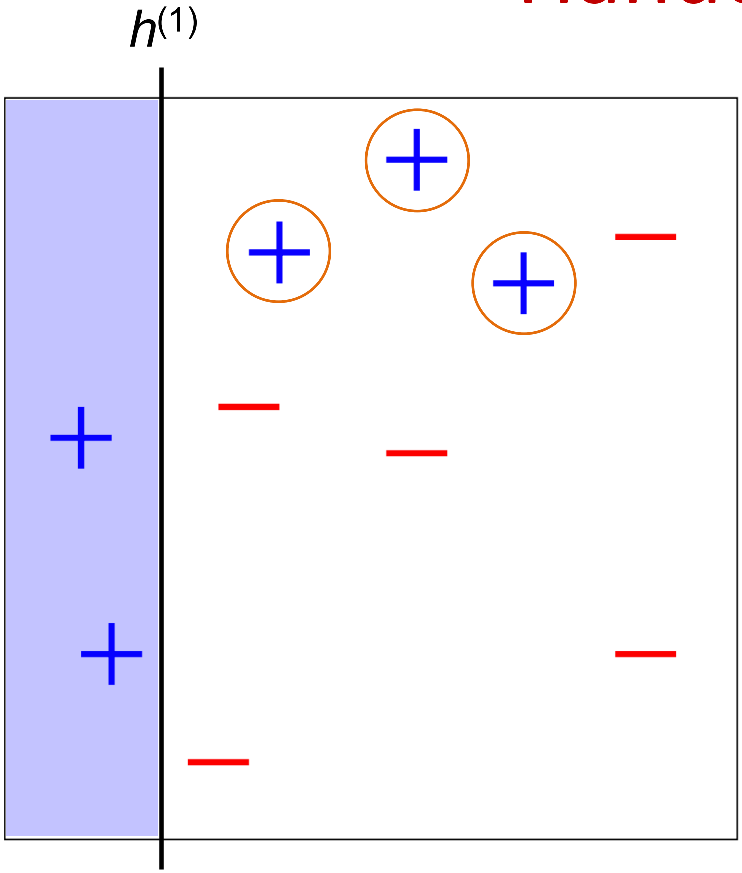
$$h(\vec{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot \underbrace{h^{(t)}(\vec{x})}_{\{-1, +1\}}\right)$$



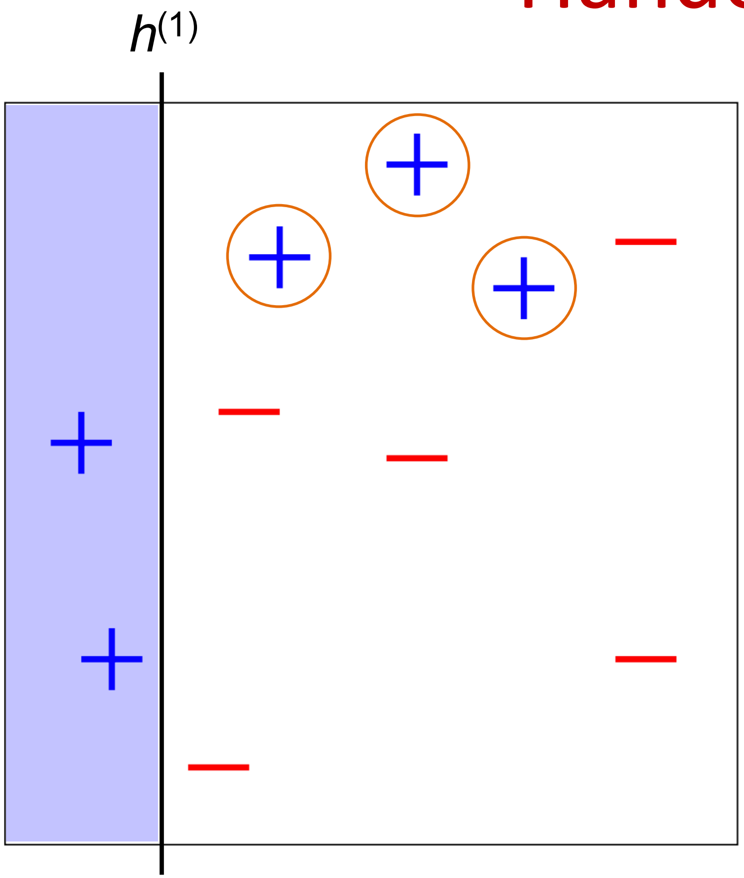
Handout 11: Round 1



Handout 11: Round 1



Handout 11: Round 1



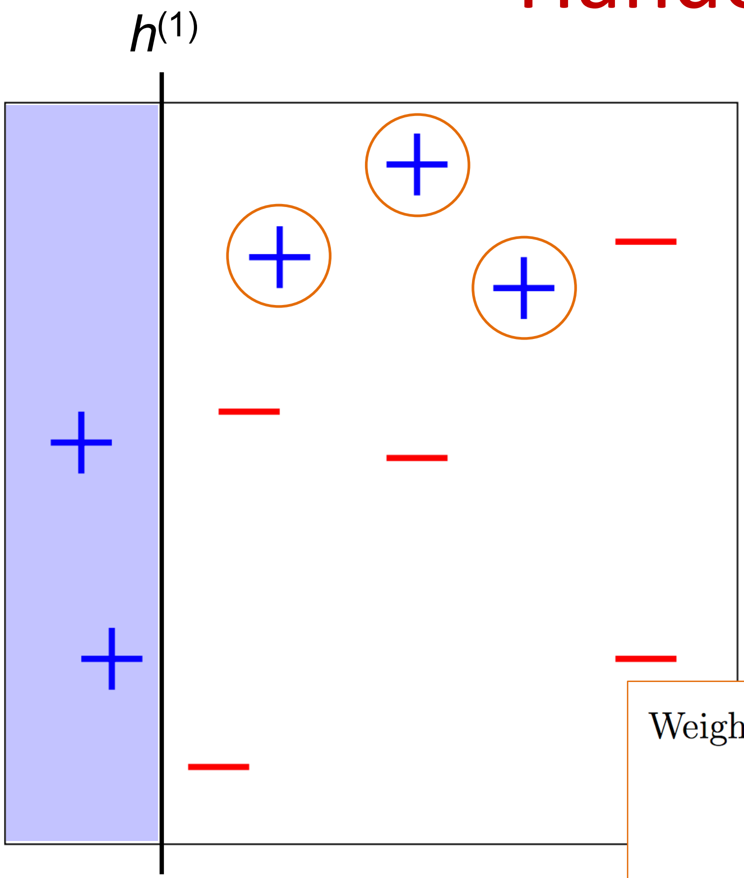
$w_i^{(1)} = \frac{1}{10}$ for all $i = 1, 2, \cdots, 10$.

$\epsilon_1 = \frac{3}{10}$ (three points incorrectly classified, all with weight $\frac{1}{10}$)

$\alpha_1 = \frac{1}{2} \ln\left(\frac{1-\frac{3}{10}}{\frac{3}{10}}\right) = \ln\sqrt{\frac{7}{3}} \approx 0.42$

- correctly classified: $w_i^{(2)} = c_1 \cdot \frac{1}{10} \exp\left(-\ln\sqrt{\frac{7}{3}}\right)$
- incorrectly classified: $w_i^{(2)} = c_1 \cdot \frac{1}{10} \exp\left(\ln\sqrt{\frac{7}{3}}\right)$

Handout 11: Round 1



$w_i^{(1)} = \frac{1}{10}$ for all $i = 1, 2, \cdots, 10$.

$\epsilon_1 = \frac{3}{10}$ (three points incorrectly classified, all with weight $\frac{1}{10}$)

$\alpha_1 = \frac{1}{2} \ln \left(\frac{1 - \frac{3}{10}}{\frac{3}{10}} \right) = \ln \sqrt{\frac{7}{3}} \approx 0.42$

- correctly classified: $w_i^{(2)} = c_1 \cdot \frac{1}{10} \exp\left(-\ln \sqrt{\frac{7}{3}}\right)$
- incorrectly classified: $w_i^{(2)} = c_1 \cdot \frac{1}{10} \exp\left(\ln \sqrt{\frac{7}{3}}\right)$

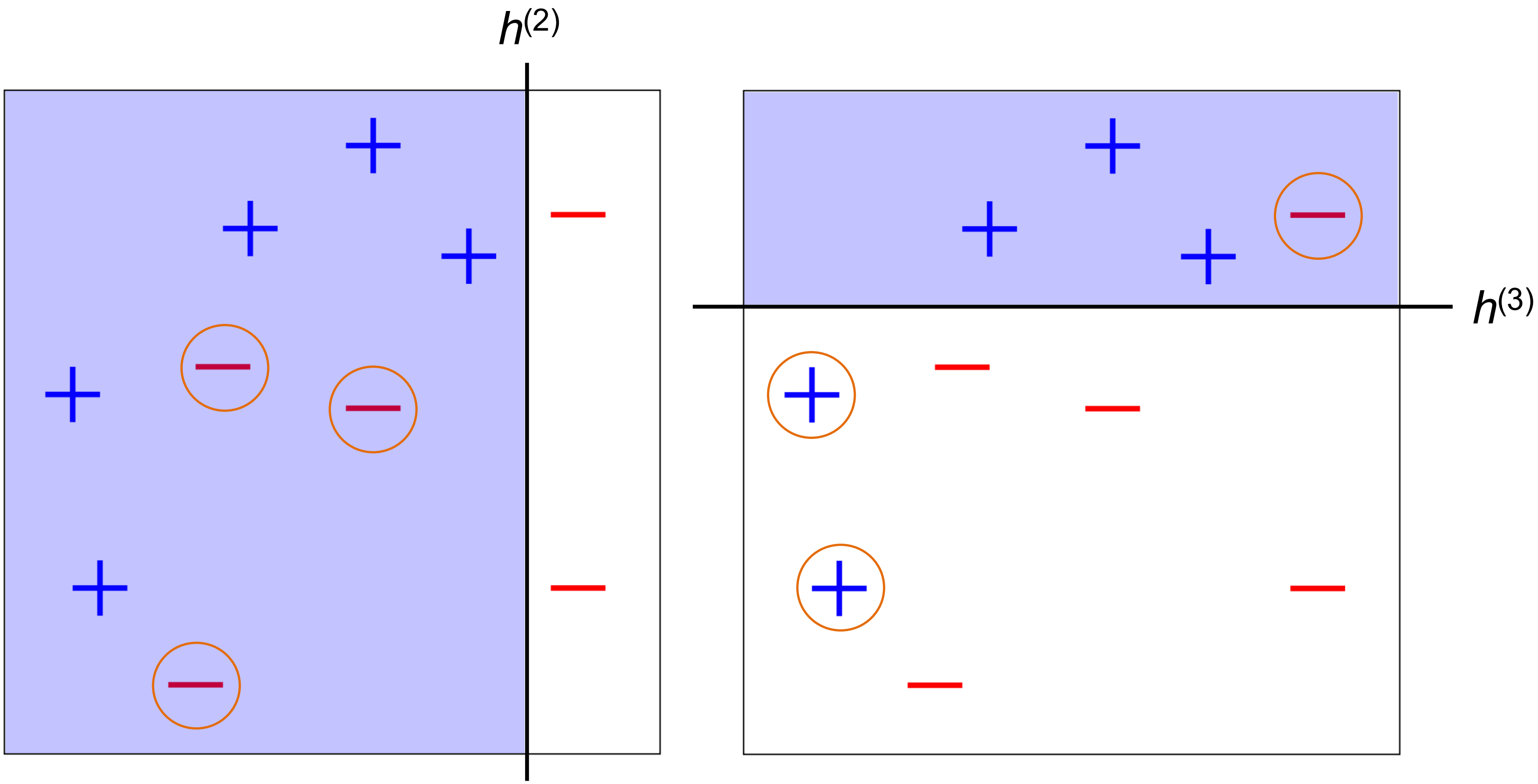
Weights must sum to 1, $\Rightarrow$

$$7 \cdot \frac{c_1}{10} \exp\left(-\ln \sqrt{\frac{7}{3}}\right) + 3 \cdot c_1 \cdot \frac{1}{10} \exp\left(\ln \sqrt{\frac{7}{3}}\right) = 1$$

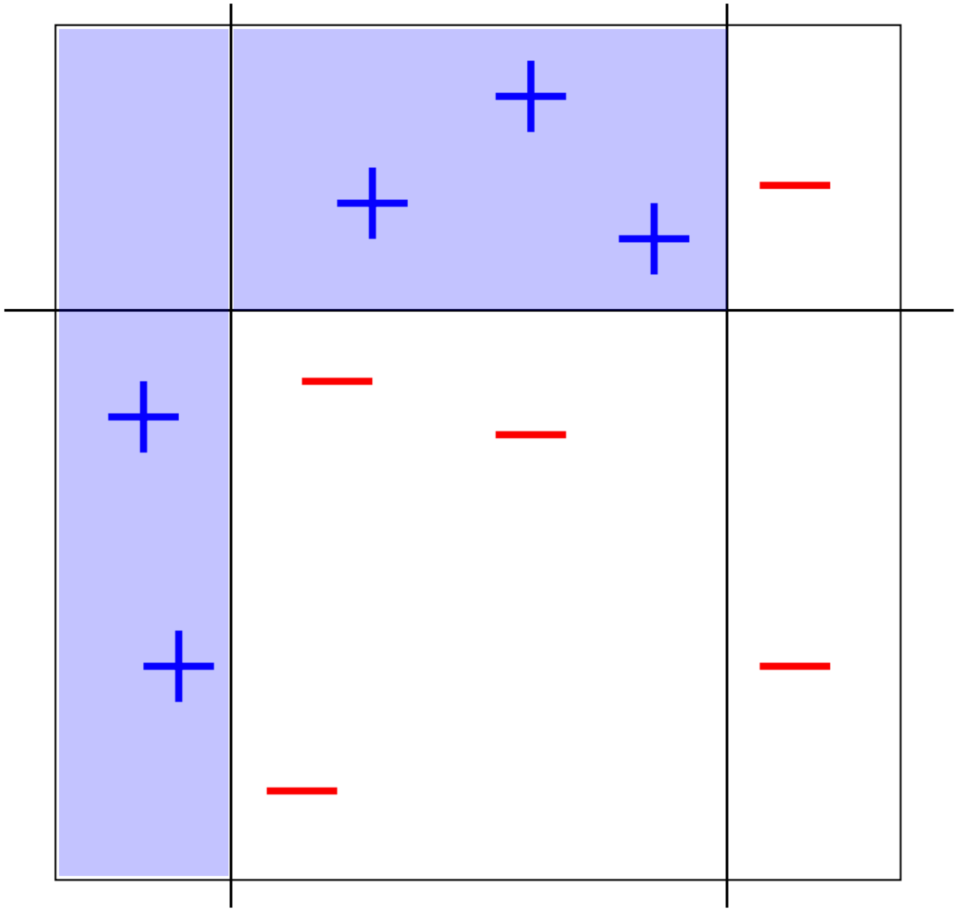
$$\Rightarrow \quad c_1 = \frac{5}{\sqrt{21}}$$

- correctly classified: $w_i^{(2)} = \frac{5}{\sqrt{21}} \cdot \frac{1}{10} \sqrt{\frac{3}{7}} = \frac{1}{14}$ decrease!
- incorrectly classified: $w_i^{(2)} = \frac{5}{\sqrt{21}} \cdot \frac{1}{10} \sqrt{\frac{7}{3}} = \frac{1}{6}$ increase!

Handout 11: Round 2 & 3 (exercise!)



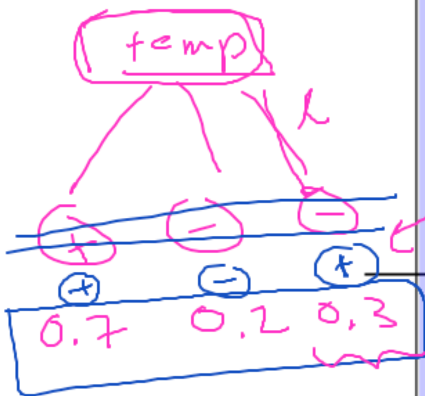
Handout 11: final classifier



$$h(\boldsymbol{x}) = \text{sign}\left(0.42 \cdot h^{(1)}(\boldsymbol{x}) + 0.65 \cdot h^{(2)}(\boldsymbol{x}) + 0.92 \cdot h^{(3)}(\boldsymbol{x})\right)$$

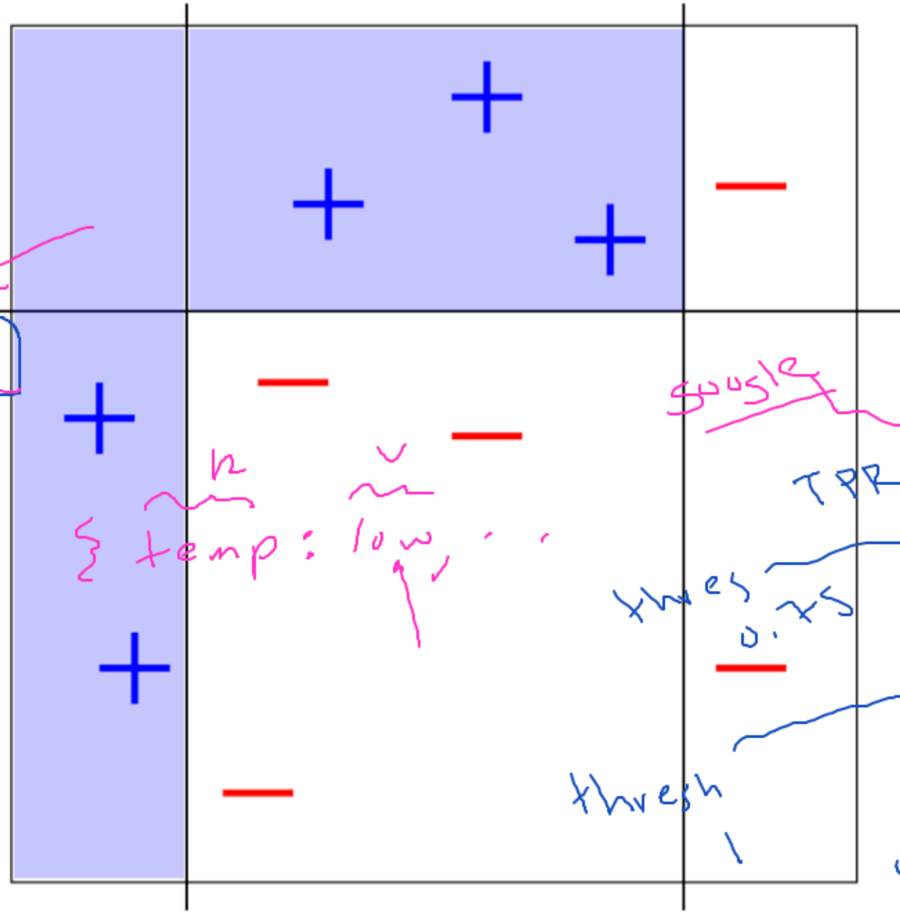
Handout 11: final classifier

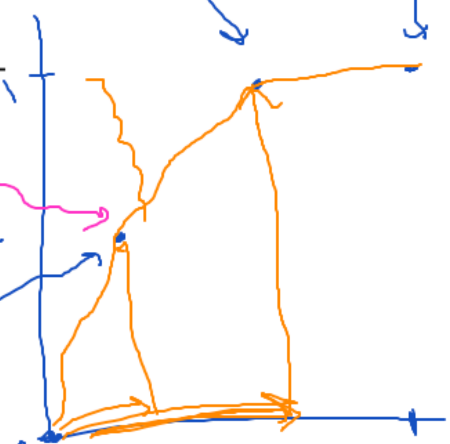
thresh = 0.25



sign$(-0.42 + 0.65 - 0.92)$

$\boxed{h(x) = -1}$





$$h(\boldsymbol{x}) = \operatorname{sign}\left(\overset{\alpha_1}{0.42} \cdot \underset{-1}{h^{(1)}(\boldsymbol{x})} + \overset{\alpha_2}{0.65} \cdot \underset{+1}{h^{(2)}(\boldsymbol{x})} + \overset{\alpha_3}{0.92} \cdot \underset{-1}{h^{(3)}(\boldsymbol{x})}\right)$$

Outline for October 27

- Recap Random Forests
- AdaBoost and weighted entropy
- Perceptron Algorithm

Next time!