Pattern Classification

07. Classifiers Combination

AbdElMoniem Bayoumi, PhD

Fall 2021

Classifiers Combination

	Classifier 1	Classifier 2	Classifier 3
$P_{correct}$	93%	92%	90%

- Robust approach
- Best classifier for the training set might not be the best for the test set
- It is risky to choose the best in training

Classifiers Combination

- Diversity is a favorable feature we would like to have in the classifiers we choose to combine

- Ways to combine:
  - Majority vote
  - Average the class posterior probabilities, i.e., $\boldsymbol{P(C_i|\underline{X})}$
  - Average some sort of score function
  - Can also choose median instead of average

AdaBoost Classifier

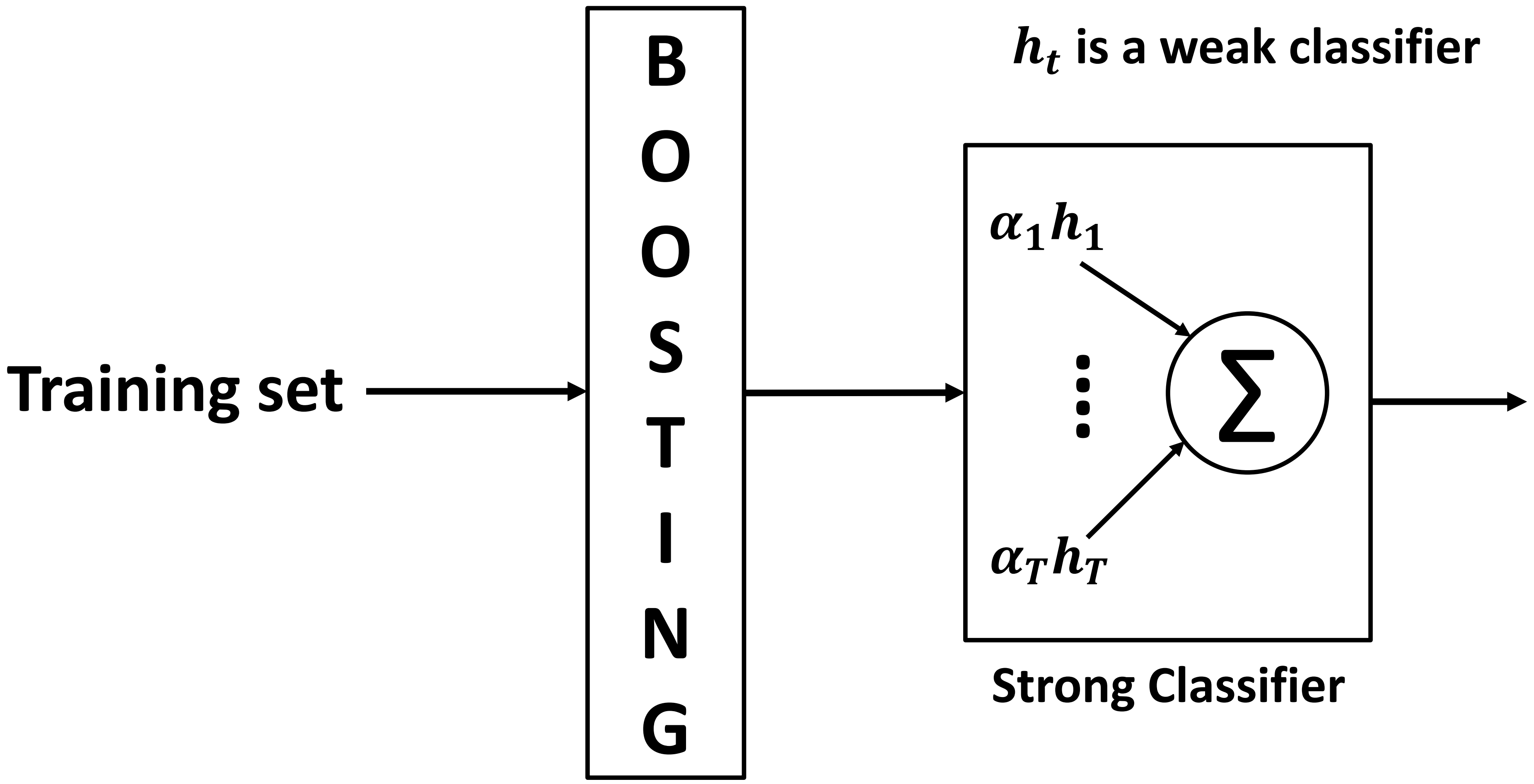
- The AdaBoost algorithm is an iterative procedure
- AdaBoost tries to approximate the Bayes classifier by combining many *weak* classifiers
- AdaBoost selects the best **weak** classifiers to create a **strong** classifier

AdaBoost Classifier

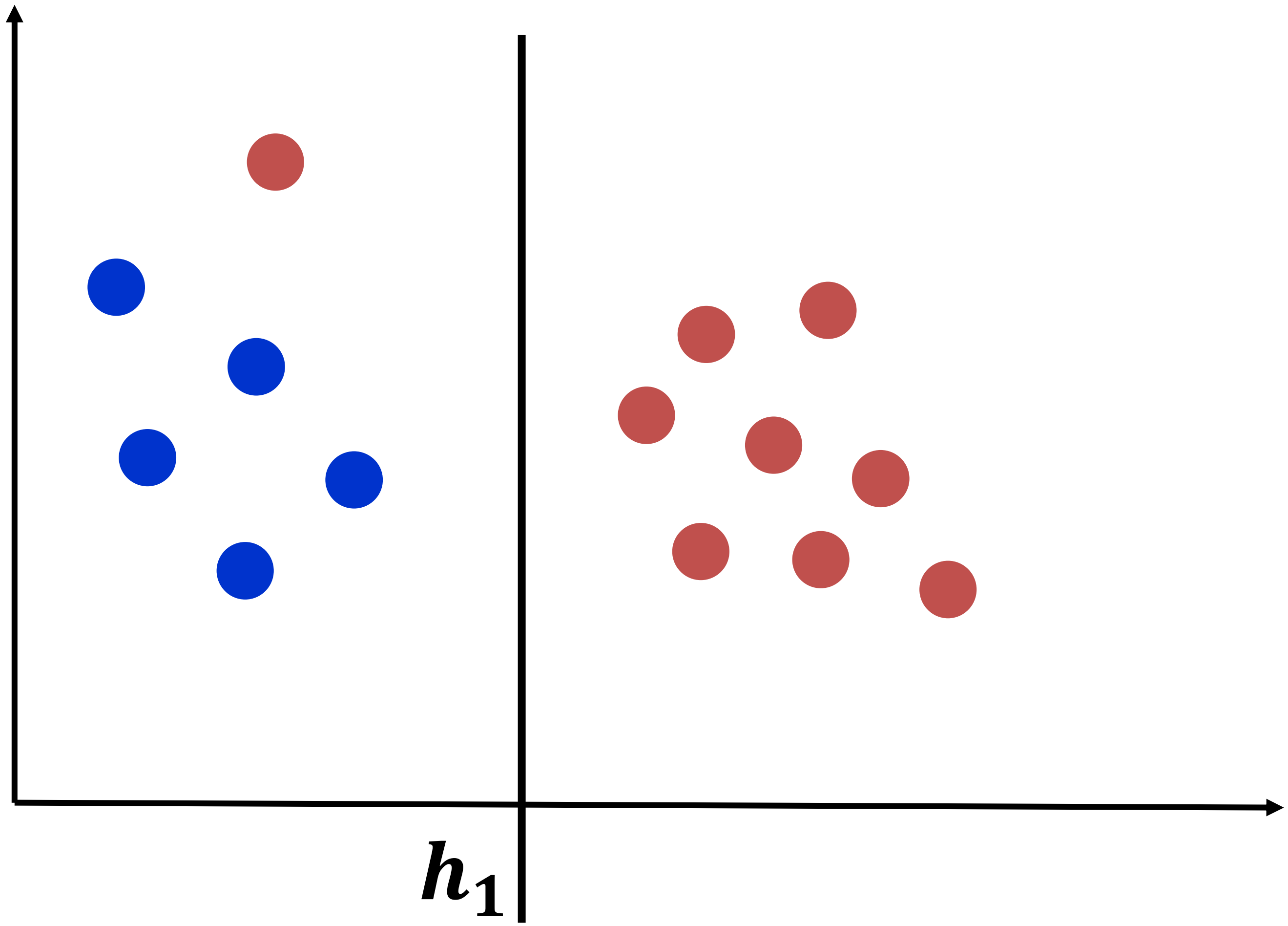
- A **weak classifier** is able to guess the right class with a percentage slightly bigger than random guessing
  - E.g., random guessing acc. is 50% in case of binary classification
  - So, a weak binary classifier’s acc. is 51% → close to random

- A **strong** classifier is usually correct > 80%

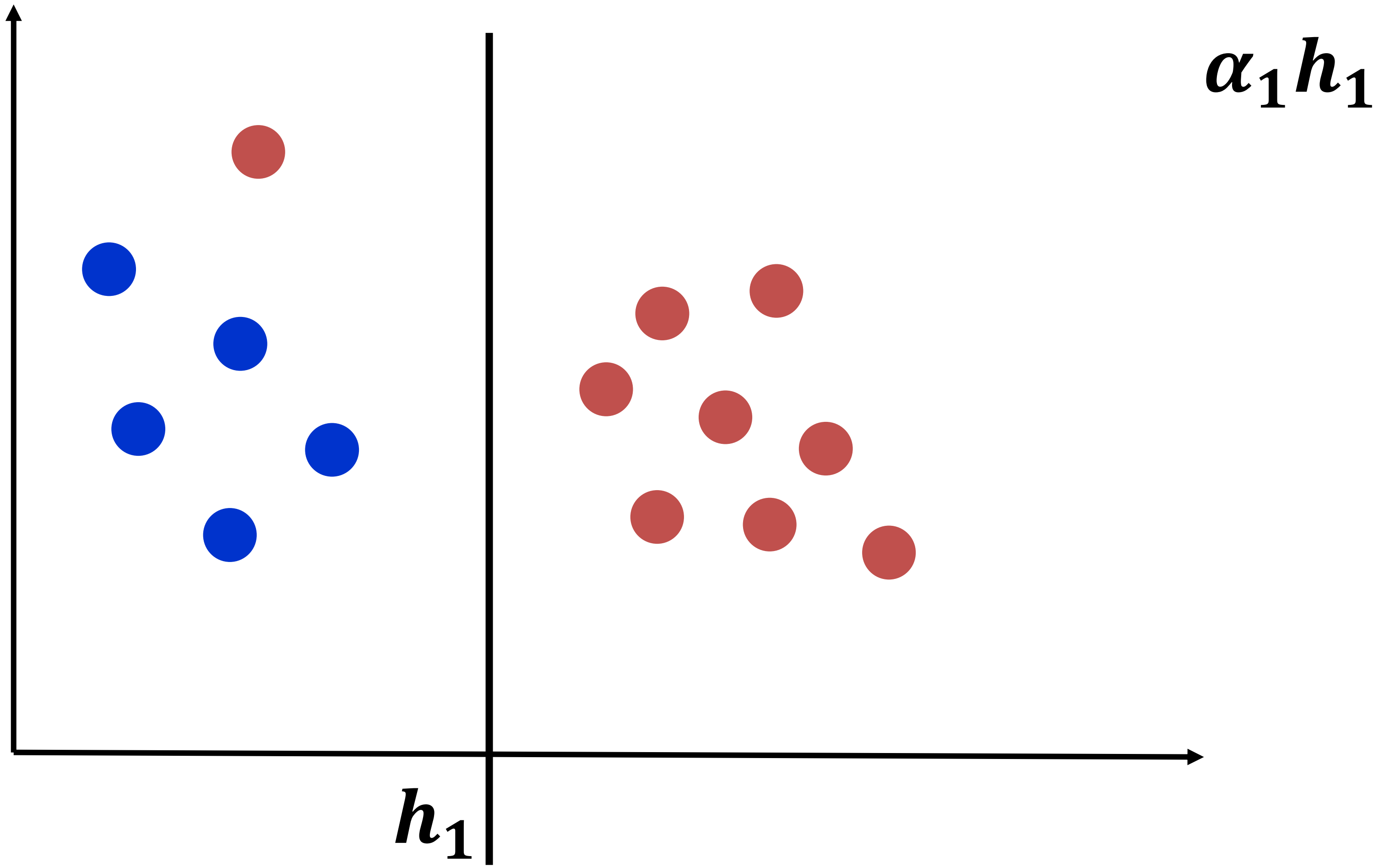
Key Idea of AdaBoost



Key Idea of AdaBoost

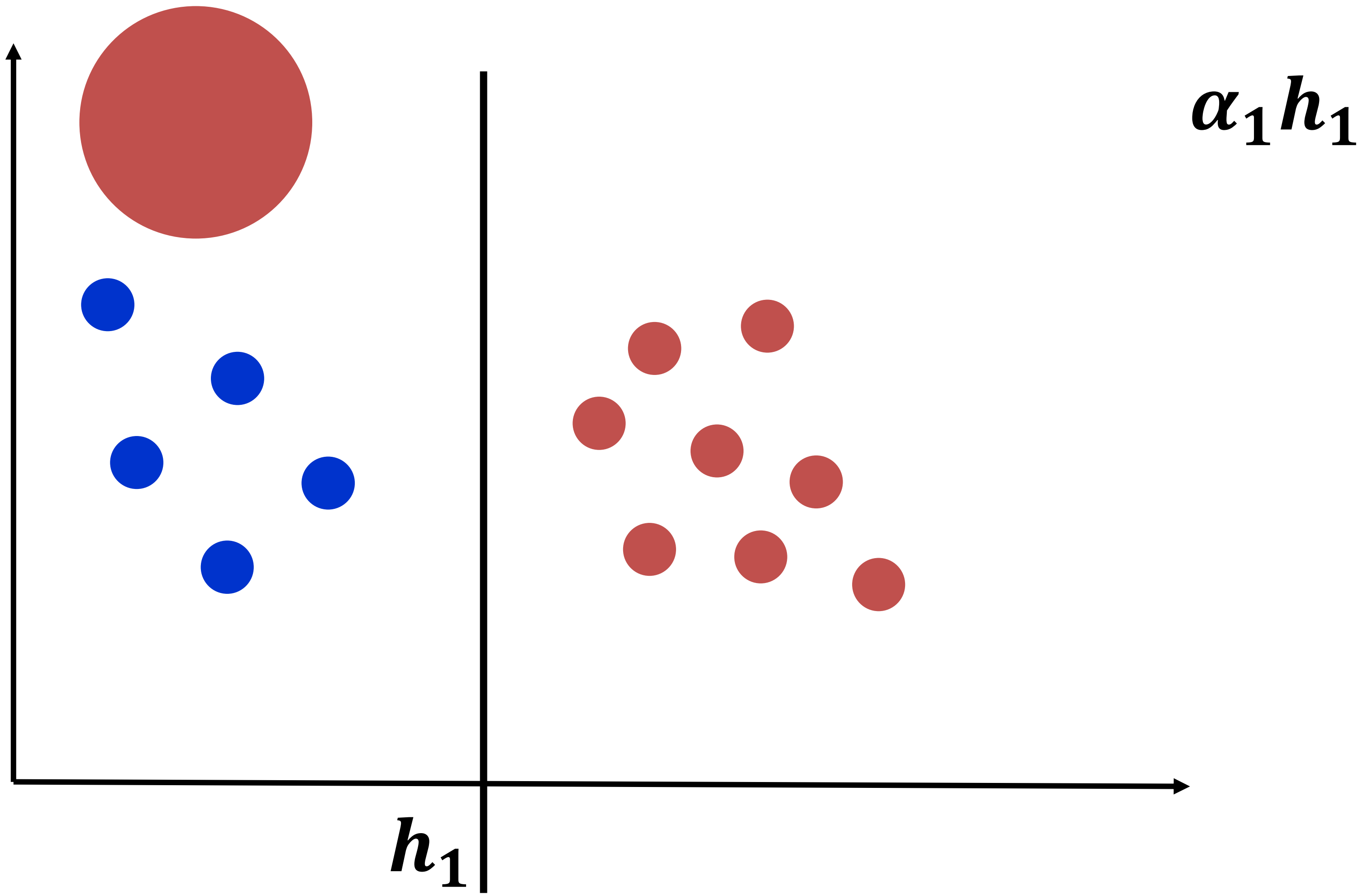


Key Idea of AdaBoost

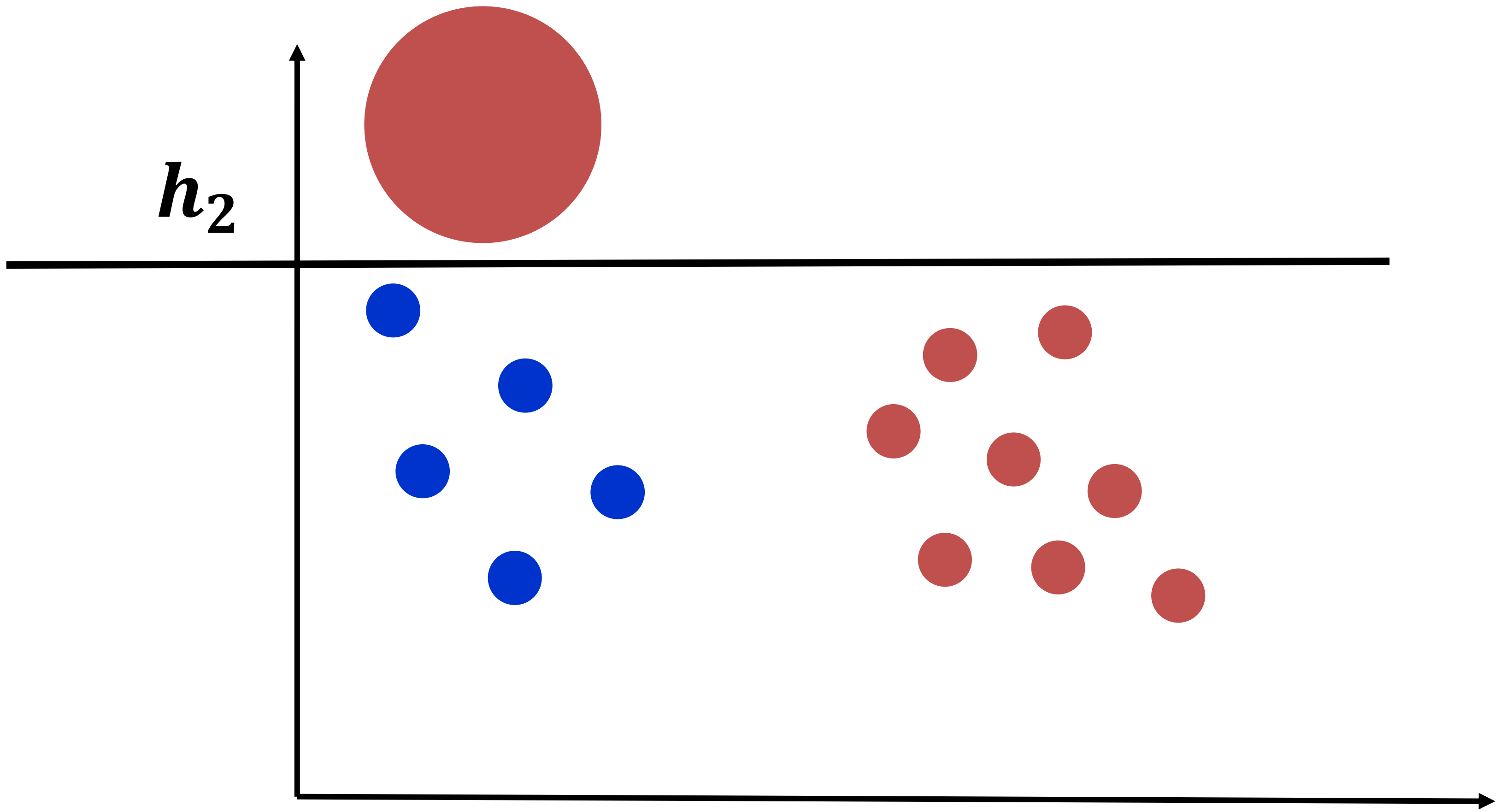


Key Idea of AdaBoost

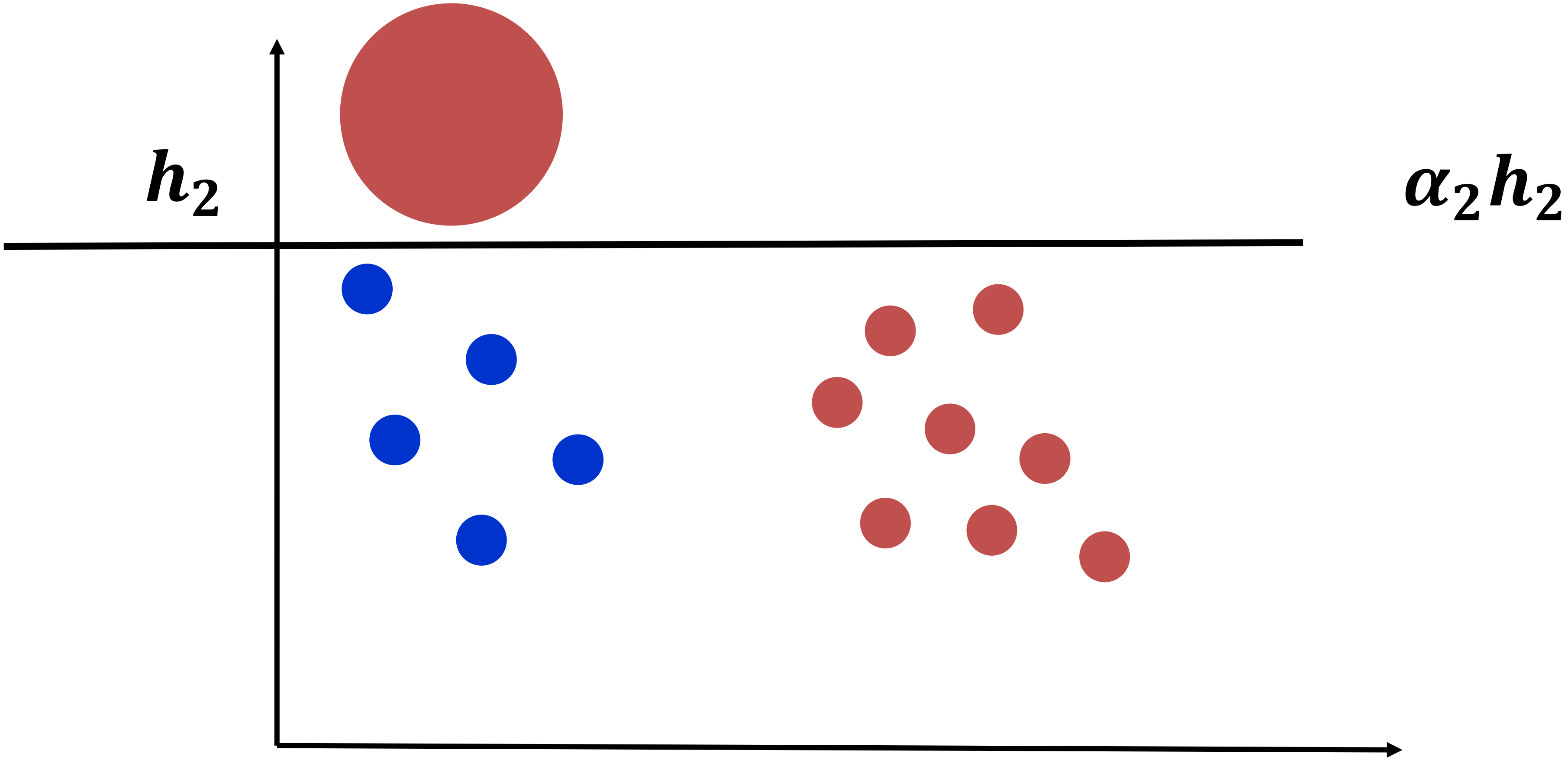
More weight for wrongly classified training examples



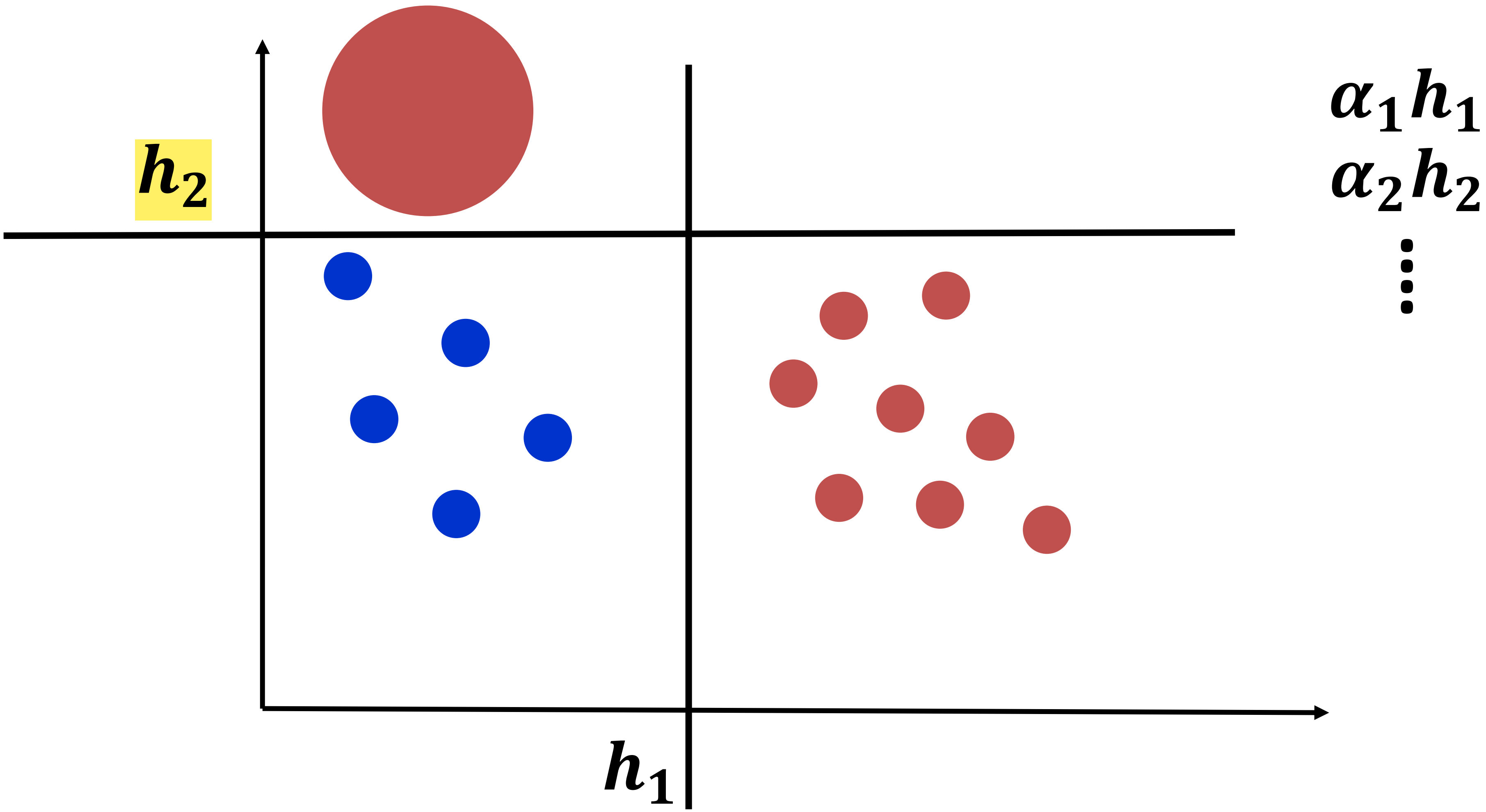
Key Idea of AdaBoost



Key Idea of AdaBoost



Key Idea of AdaBoost



AdaBoost Algorithm

$$\mathbb{I}(y) = \begin{cases} 1 & y = True \\ 0 & y = False \end{cases}$$

1. Initialize weights of the training examples:

$$w_m = \frac{1}{M} \quad , m = 1, 2, \dots, M$$

2. For t=1 to T:
   a) Select a classifier h_t that best fits to the training data using weights w_m of the training examples
   b) Compute error of h_t as: $err_t = \frac{\sum_{m=1}^{M} w_m \mathbb{I}(c_m \neq h_t(x_m))}{\sum_{m=1}^{M} w_m}$
   c) Compute weight of classifier: $\alpha_t = \log\left(\frac{1-err_t}{err_t}\right)$
   d) Update weights of wrongly classified examples:

$$w_m \leftarrow w_m \, e^{\alpha_t \, \mathbb{I}(c_m \neq h_t(x_m))} \qquad \textbf{for } m = 1 \dots M$$

   e) Renormalize weights w_m
3. Output: $C(x) = \underset{k}{\operatorname{argmax}} \sum_{t=1}^{T} \alpha_t \mathbb{I}(h_t(x) = k)$

AdaBoost Classifier

- AdaBoost works well only in case of binary classification
- However, it is not the case for multi-class classification
- AdaBoost assumes that the error of each weak classifier is less than 0.5
  - $\boldsymbol{\alpha_t}$ is negative if error is greater than 0.5
  - The error of random guessing in case of two classes is 0.5
  - In case of K multi-classes the random guessing error rate is $\frac{K-1}{K}$

Multi-Class AdaBoost

- Same as AdaBoost algorithm except for step (2.c):

$$\boldsymbol{\alpha_t = log\left(\frac{1-err_t}{err_t}\right) + \log(K-1)}$$

- $\boldsymbol{\alpha_t}$ is positive if only:

$$\boldsymbol{(1 - err_t) > \frac{1}{K}}$$

Overfitting

- AdaBoost is robust to overfitting given that select the best **weak classifiers**
- However, relying on complex classifiers will be more prone to overfitting

Acknowledgment

- These slides have been created relying on lecture notes of Amir Atiya and Sven Behnke

- J. Zhu, H. Zou, S. Rosset, T. Hastie, “Multi-class AdaBoost”, 2009