电子科技大学
University of Electronic Science and Technology of China

Ensemble learning

Fangfang Chen

Data Mining Lab, Big Data Research Center, UESTC

Email：fangfchen@126.com

Outline

1 Introduction

- Background
- Concept of ensemble learning
- Steps of ensemble learning

2 Ensemble methods

- Bagging
- Boosting
- Comparison between Bagging & Boosting

1 Introduction

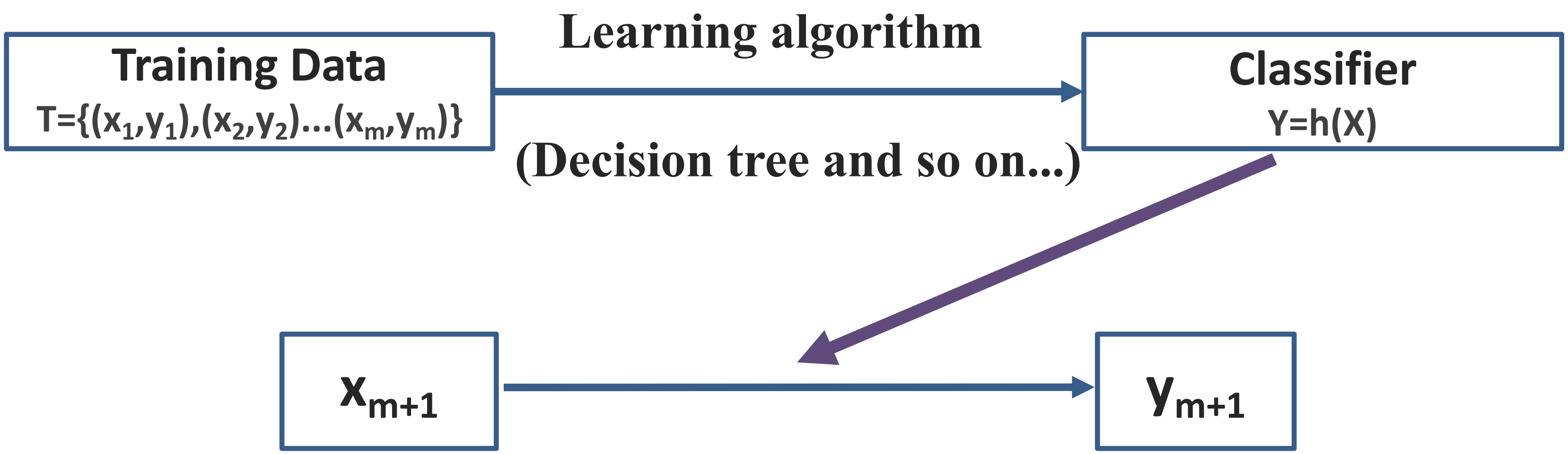
1.1 Background

1.2 Concept of ensemble learning

1.3 Steps of ensemble learning

1.1 Background

- **Classfication**

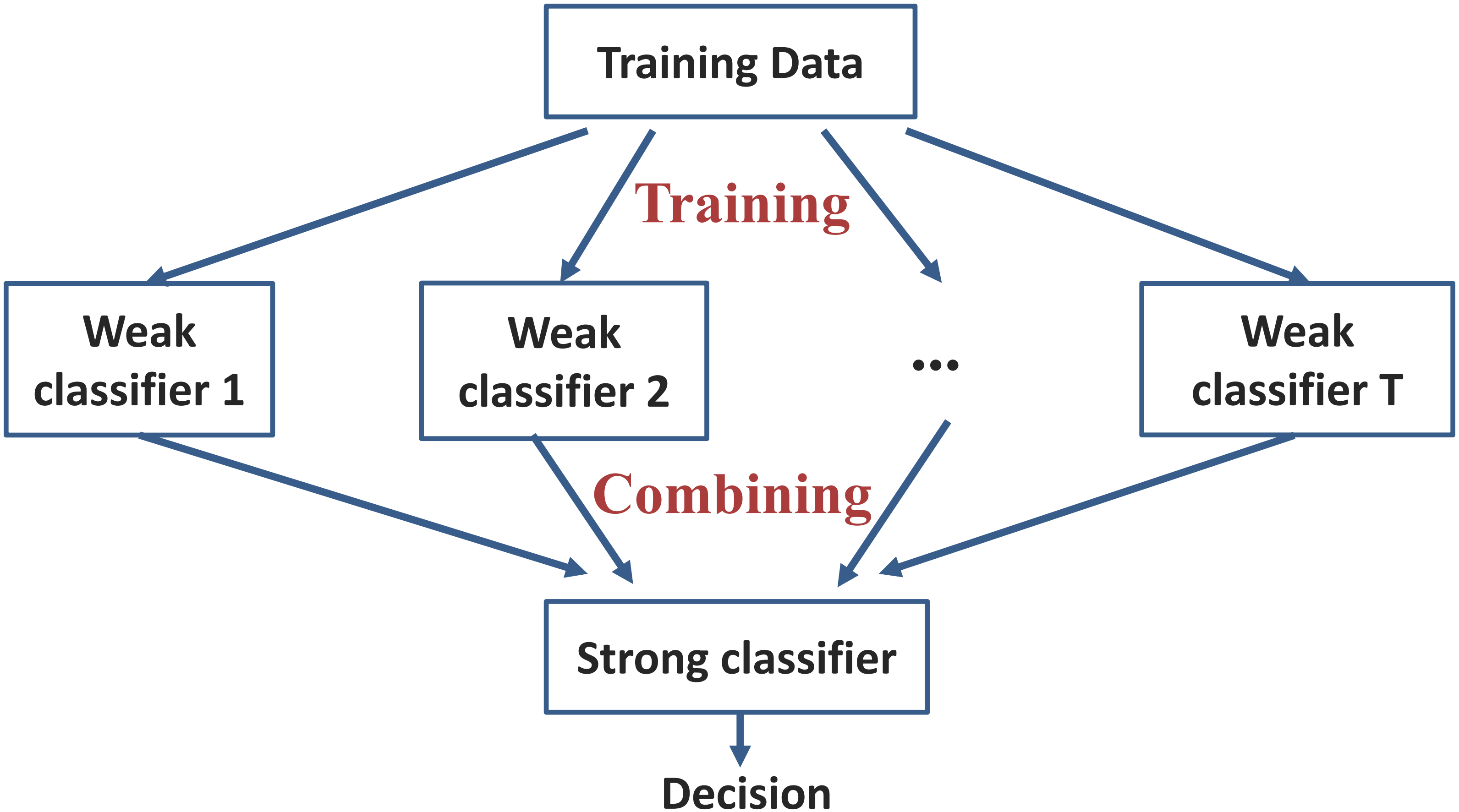


- Weak classifier — slightly better than random guess (Easy to get)
- Strong classifier — can make very accurate predictions (Hard to get!)

1.2 Concept of ensemble learning

- **Ensemble learning**

  Multiple classifiers are trained and combined to solve a same problem.



1.3 Steps of ensemble learning

数据挖掘实验室

Data Mining Lab

- Two steps:

1、Train a number of base(weak) classifiers

- Common methods — Sampling:

— Generate a number of samples according to the training data set

— Train the base classifiers from these different samples (but using the same learning algorithm)

2、Combine the base classifiers to use

- Common methods:

✓ Majority voting (e.g. Bagging)

✓ Weighted majority voting (e.g. Boosting)

2 Ensemble methods

2.1 Bagging

- Framework
- Pseudo-code
- Features

2.2 Boosting

- Framework
- Adaboost algorithm
- Pseudo-code
- A example of Adaboost
- Features

2.3 Comparison between Bagging & Boosting

2.1 Bagging

数据挖掘实验室

Data Mining Lab

- **Framework**

1、Training

— Sample the training data set randomly with replacement (有放回抽取)

— The size of a sample is as the same as that of the training data set generally

2、Combining

— Majority voting

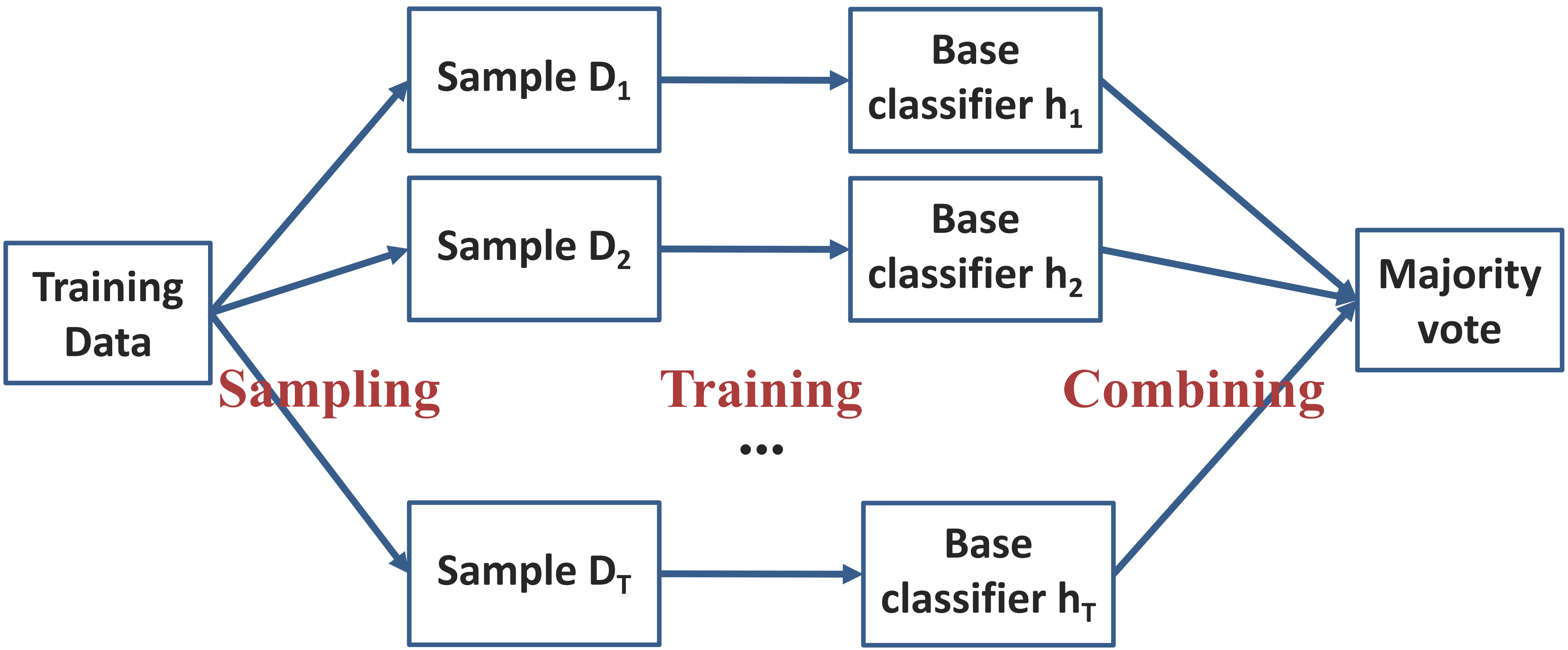
— The most-voted class is predicted

2.1 Bagging

数据挖掘实验室

Data Mining Lab

- Framework



2.1 Bagging

数据挖掘实验室
Data Mining Lab

- **Pseudo-code**

Input: Data set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
Base learning algorithm $\mathcal{L}$;
Number of learning rounds T.

The number of base classifiers

Process:
for $t = 1, \cdots, T$:

Complete name of bagging is **Bootstrap aggregating**

$\mathcal{D}_t = Bootstrap(\mathcal{D})$; % Generate a bootstrap sample from $\mathcal{D}$
$h_t = \mathcal{L}(\mathcal{D}_t)$ % Train a base learner h_t from the bootstrap sample
end.

Output: $H(\boldsymbol{x}) = \text{argmax}_{y \in \mathcal{Y}} \sum_{t=1}^{T} 1(y = h_t(\boldsymbol{x}))$ % the value of $1(a)$ is 1 if a is $true$ and 0 otherwise

---

Find the most-voted class

2.1 Bagging

- **Features**

  ✓ Samples are independent

  ✓ Base classifiers can be generated in a parallel style
  — Save time

  ✓ For unstable learning algorithm bagging works well
  (e.g. Decision tree, neural network)

2.2 Boosting

- **Framework**

- **Main idea:**

  Learn the examples with high error rate intensively

1、Training

— Assign **equal weights** (probabilities) to all the training examples

— Train a base classifier from the training data set

— Test it , and update the weights (**Increasing the weights of incorrectly classified examples**)

— Train next classifier from **updated weight distribution**(consider more about incorrect examples), repeat for T times

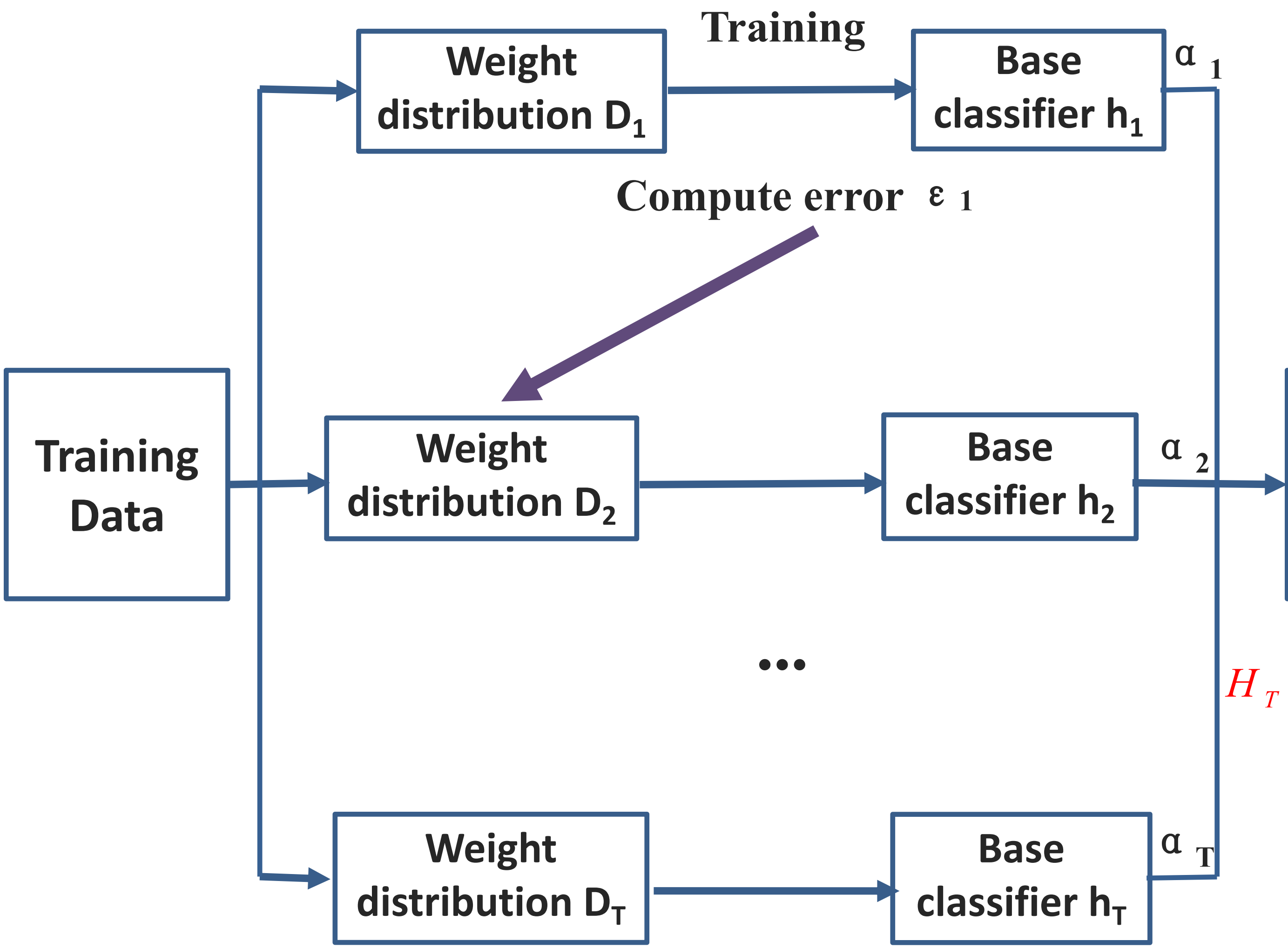
- Sample according to the weight distribution if needed

2、Combining

— **Weighted** majority voting (linear combination)

2.2 Boosting

- Framework



$$H_T(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

- **Adaboost algorithm**

Given:

$$(x_1, y_1), \ldots, (x_m, y_m) \text{ where } x_i \in X, y_i \in \{-1, +1\}$$

(1) Initialization:

$$D_1(i) = \tfrac{1}{m}, i = 1, \ldots, m$$

- Assign **equal weights** (probabilities) to all the training examples
- The sum of weights is equal to 1 — a distribution

2.2 Boosting

- **Adaboost algorithm**

(2) Training: $t = 1, \ldots, T$

a. Generate a classifier which minimizes error

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

$$h_t = \underset{h_j}{\arg\min}\, \varepsilon_j$$

b. Measure the error of h_t

$$\varepsilon_t = \sum_{i=1}^{m} D_t(i)[y_i \neq h_t(x_i)]$$

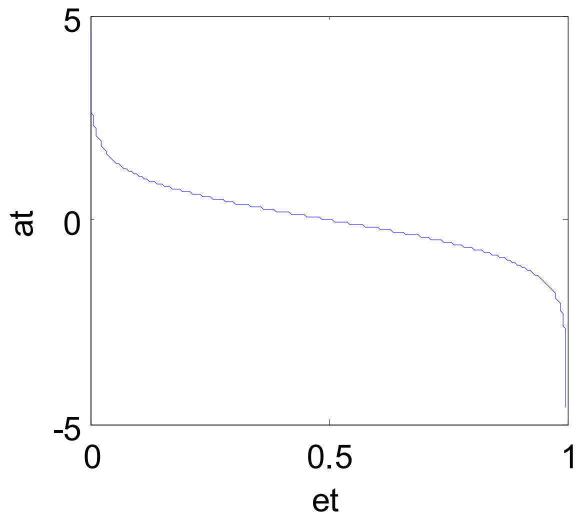
- **According to the weight distribution D_t**
- **Reflect the effect of weights**

- **Adaboost algorithm**

(2) Training: $t = 1, \dots, T$

c. Determine the weight of classifier h_t

$$\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$$



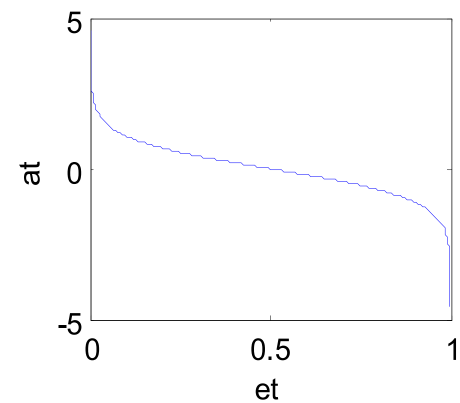
- **Denote the significance or reliability**
- **Monotone decreasing**
- **When $\varepsilon_t < 1/2$, $\alpha_t > 0$; when $\varepsilon_t > 1/2$, $\alpha_t < 0$**

- **Adaboost algorithm**

(2) Training: $t = 1, \ldots, T$

$$\alpha_t = \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t}$$

d. Update distribution



$$D_{t+1}(i) = \frac{D_t(i)\exp[-\alpha_t y_i h_t(x_i)]}{Z_t},\ Z_t \text{ is for normalization}$$

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

- **The weight of incorrect example amplified by $e^{-\alpha t}$ than correct example**

- **The weight is increased when**

$$\alpha_t > 0(\varepsilon_t < \frac{1}{2}) \quad \text{and} \quad h_t(x_i) \neq y_i$$

or

$$\alpha_t < 0(\varepsilon_t > \frac{1}{2}) \quad \text{and} \quad h_t(x_i) = y_i$$

2.2 Boosting

- **Adaboost algorithm**

(3) Combining:

$$sign\left(H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \right)$$

- Sign of H(x) — the result of classification
- Absolute value of H(x) — the reliability of classification
- The sum of αt is not equal to 1

2.2 Boosting

数据挖掘实验室
Data Mining Lab

- **Pseudo-code**

Input: Data set $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$;
Base learning algorithm $\mathcal{L}$;
Number of learning rounds T.

Process:

$D_1(i) = 1/m$. % Initialize the weight distribution

for $t = 1, \cdots, T$:

$h_t = \mathcal{L}(\mathcal{D}, D_t)$; % Train a base learner h_t from $\mathcal{D}$ using distribution D_t

$\epsilon_t = \Pr_{i \sim D_i}[h_t(\boldsymbol{x}_i \neq y_i)]$; % Measure the error of h_t

$\alpha_t = \frac{1}{2}\ln \frac{1-\epsilon_t}{\epsilon_t}$; % Determine the weight of h_t

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

$$= \frac{D_t(i)\exp(-\alpha_t y_i h_t(\boldsymbol{x}_i))}{Z_t}$$

% Update the distribution, where Z_t is a normalization
% factor which enables D_{t+1} to be a distribution

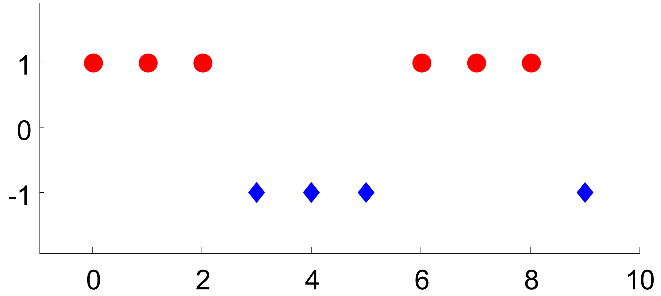
end.

Output: $H(\boldsymbol{x}) = \text{sign}\left(f\left(\boldsymbol{x}\right)\right) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})\right)$

2.2 Boosting

- **A example of Adaboost**
  - **Training Data:**

No.	1	2	3	4	5	6	7	8	9	10
x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1



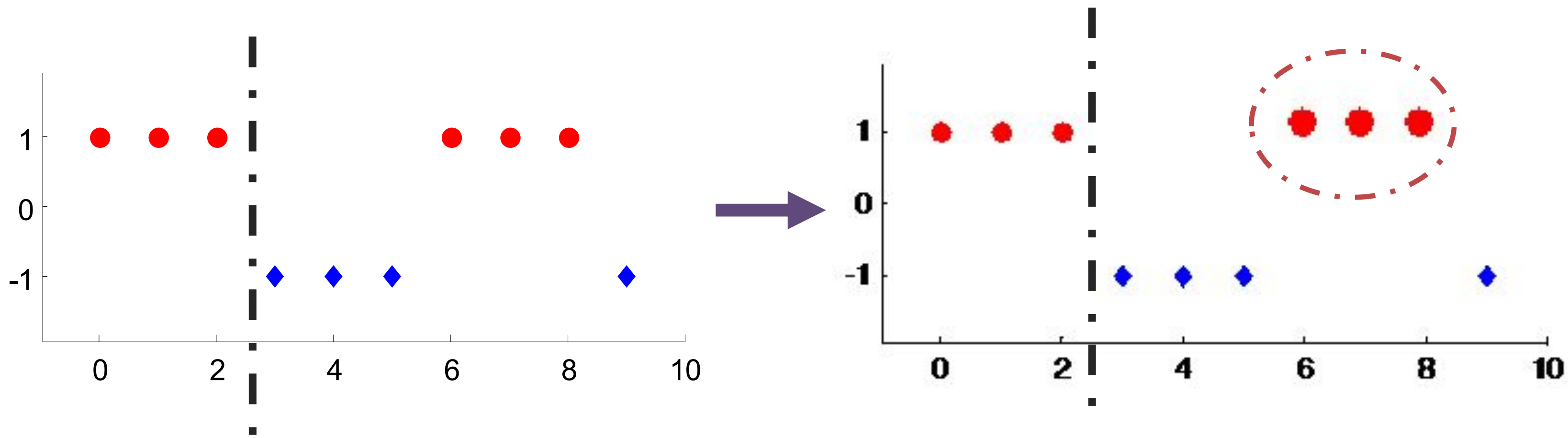
  - **Initialization:**

$$D_1 = (w_{11}, w_{12}, \ldots, w_{110})$$

$$w_{1i} = 0.1, i = 1, 2, \ldots, 10$$

2.2 Boosting

- **A example of Adaboost**
  - **t=1:**

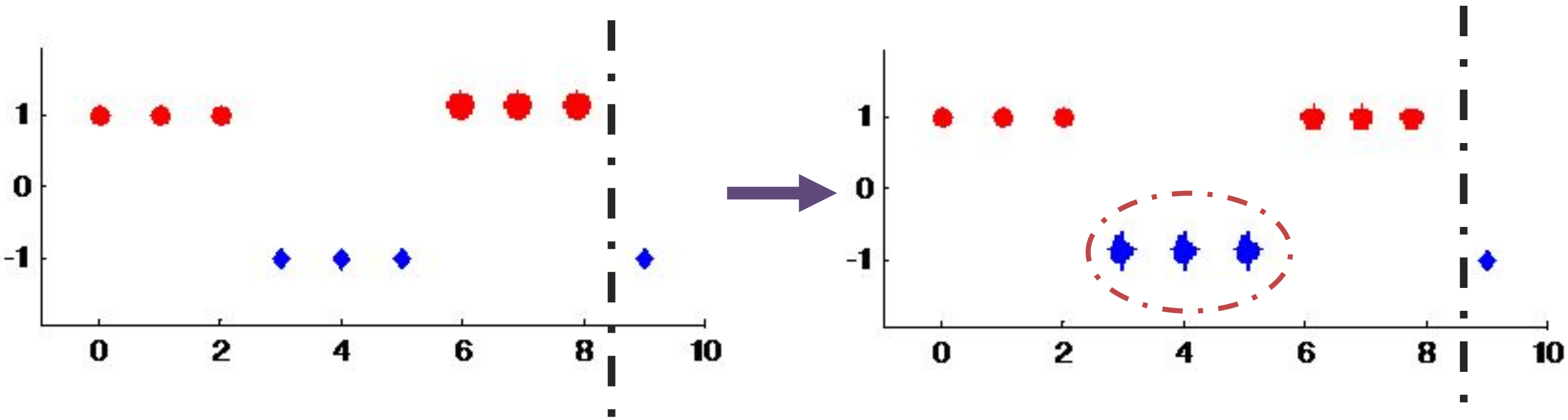


$$h_1(x) = \begin{cases} 1, x < 2.5 \\ -1, x > 2.5 \end{cases} \qquad \varepsilon_1 = 0.3 \quad \alpha_1 = 0.4236$$

$$D_2 = (0.071, 0.071, 0.071, 0.071, 0.071, 0.071, 0.167, 0.167, 0.167, 0.071)$$

2.2 Boosting

- **A example of Adaboost**
  - **t=2:**

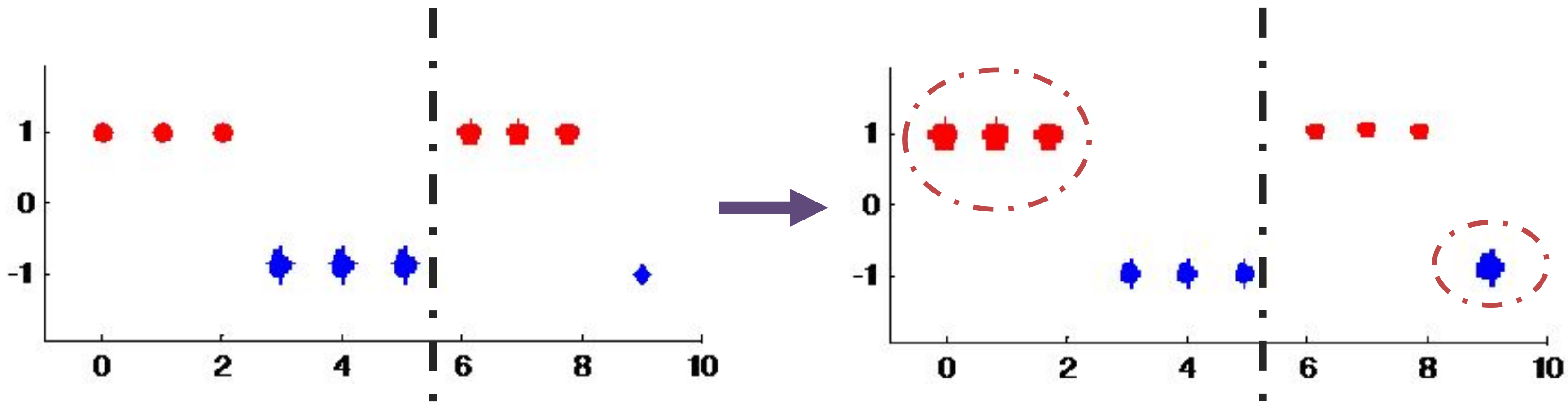


$$h_2(x) = \begin{cases} 1, x < 8.5 \\ -1, x > 8.5 \end{cases} \qquad \varepsilon_2 = 0.2143 \quad \alpha_2 = 0.6496$$

$$D_3 = (0.046, 0.046, 0.046, 0.167, 0.167, 0.167, 0.106, 0.106, 0.106, 0.046)$$

2.2 Boosting

- **A example of Adaboost**
  - **t=3:**



$$h_3(x) = \begin{cases} 1, x > 5.5 \\ -1, x < 5.5 \end{cases} \qquad \varepsilon_3 = 0.182 \quad \alpha_3 = 0.7514$$

$$D_3 = (0.125, 0.125, 0.125, 0.102, 0.102, 0.102, 0.065, 0.065, 0.065, 0.125)$$

2.2 Boosting

数据挖掘实验室

Data Mining Lab

- **A example of Adaboost**

- **Combining:**

$$h_1(x) = \begin{cases} 1, x < 2.5 \\ -1, x > 2.5 \end{cases} \qquad h_2(x) = \begin{cases} 1, x < 8.5 \\ -1, x > 8.5 \end{cases} \qquad h_3(x) = \begin{cases} 1, x > 5.5 \\ -1, x < 5.5 \end{cases}$$

$$\Downarrow$$

$$H(x) = \mathrm{sign}[0.4236h_1(x) + 0.6496h_2(x) + 0.7514h_3(x)]$$

- **The number of incorrectly classified examples is 0.**

2.2 Boosting

数据挖掘实验室
Data Mining Lab

- **Features**

✓ Samples are not independent

✓ Base classifiers should be generated in a sequential style (the generation of a base classifier has influence on the generation of subsequent classifiers)

✓ Empirical observations show that Boosting often does not suffer from overfitting even after a large number of rounds (but overfitting may occur on some special training data)

2.3 Comparison between Bagging & Boosting

	Bagging	Boosting
Sampling	Ramdomly Independent	According to the error Not independent
Generating style	Parallel	Sequential
Combining method	Majority voting	Weighted majority voting
Performance	Better than single classifier	Better than Bagging

Thanks

Fangfang Chen
Yingcai Experimental School
fangfchen@126.com