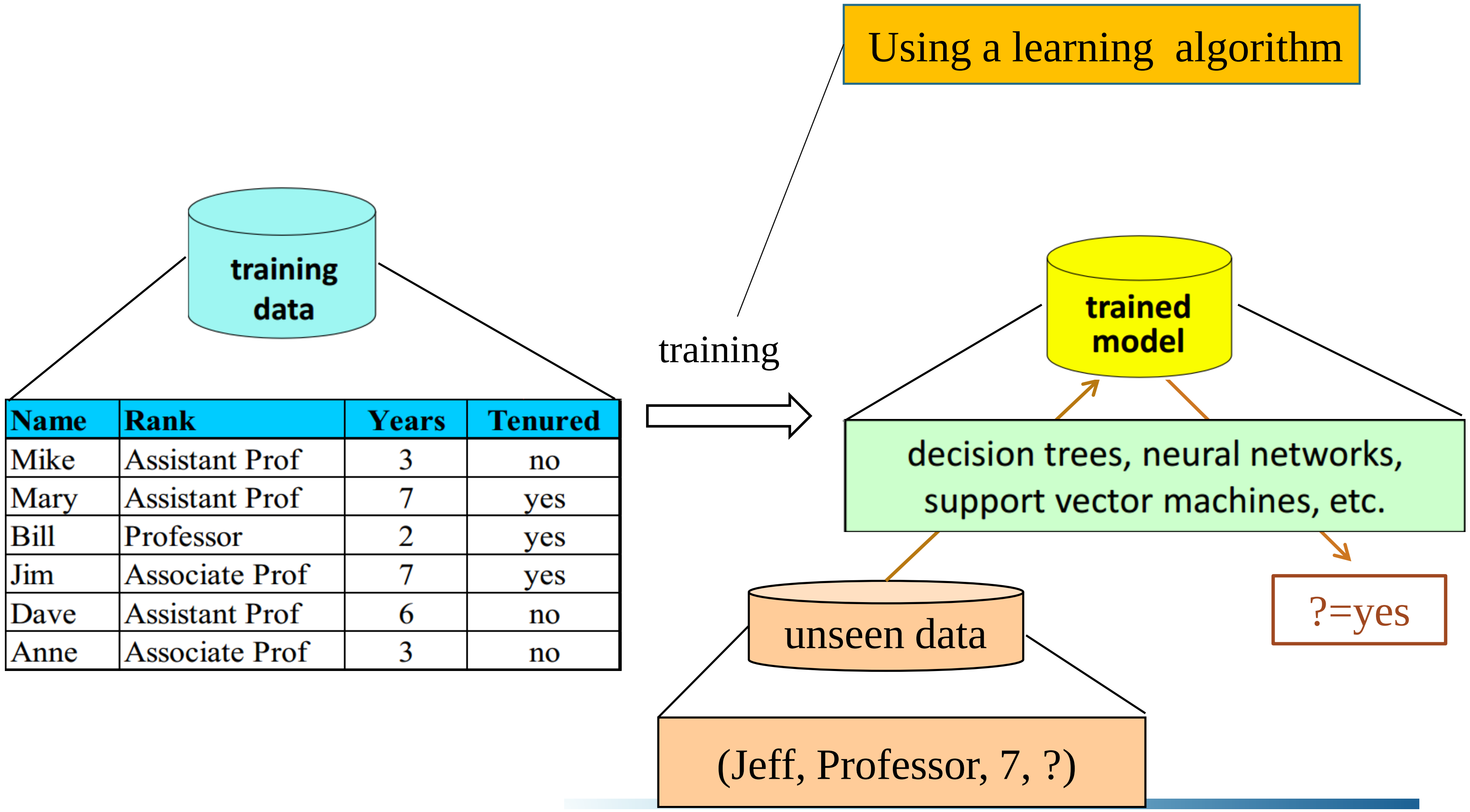
Ensemble Learning

A typical mahcine learning process


Using a learning algorithm
training data
training
trained model
decision trees, neural networks, support vector machines, etc.
unseen data
?=yes
(Jeff, Professor, 7, ?)

Name	Rank	Years	Tenured
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

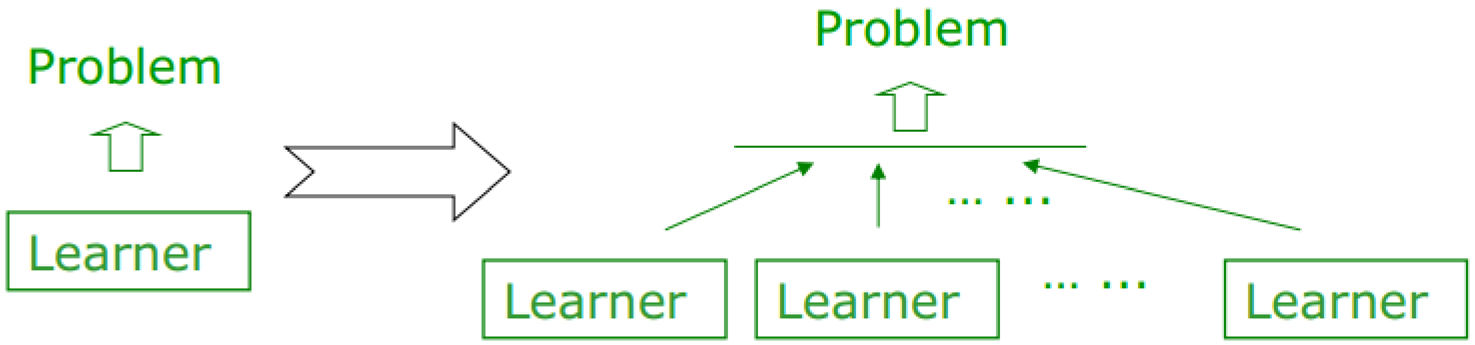
Outline

- Ensemble learning
- Ensemble method
  - Boosting
  - Bagging
- Fusion strategy
  - Averaging
  - Voting
  - Stacking
- Diversity

Ensemble Learning

"Ensemble Learning" is a machine learning paradigm where multiple (homogenous/heterogeneous) individual leaners are generated and combined for the same problem.

e.g. neural network ensemble, decision tree ensemble, etc.



Ensemble Learning

- ➢homogeneous ensemble(同质集成)

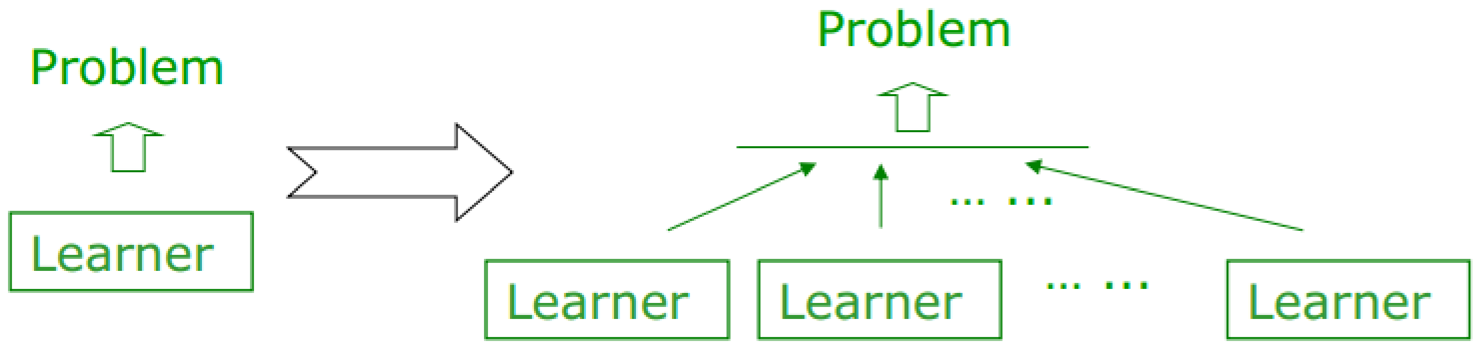
  All individual learners are genereated by the same model

- ➢heterogenous ensemble(异质集成)

  All individual learners are genereated by different model

- ➢ weak learner

  The generalization performance of learner is better than the random guessing.



Great success of ensemble methods

- KDDCup'11: 1st place of Track 1 for "A Linear Ensemble … "; 2nd place of Track 1 for "Collaborative filtering Ensemble", 1st place of Track 2 for "Ensemble …"; 2nd place of Track 2 for "Linear combination of …"
- KDDCup'12: 1st place of Track 1 for "Combining… Additive Forest…"; 1st place of Track 2 for "A Two-stage Ensemble of…"
- KDDCup'13: 1st place of Track 1 for "Weighted Average Ensemble"; 2nd place of Track 1 for "Gradient Boosting Machine"; 1st place of Track 2 for "Ensemble the Predictions"

Great success of ensemble methods

- ◻ KDDCup'14: 1st place for "ensemble of GBM, ExtraTrees, Random Forest…" and "the weighted average"; 2nd place for "use both R and Python GBMs"; 3rd place for "gradient boosting machines... random forests" and "the weighted average of…"

- ◻ Netflix Prize:
  - ✓ 2007 Progress Prize Winner: Ensemble
  - ✓ 2008 Progress Prize Winner: Ensemble
  - ✓ 2009 $1 Million Grand Prize Winner:

    Ensemble !!

An Example

	test 1	test 2	test 3
h1	√	√	×
h2	×	√	√
h3	√	×	√
	√	√	√

(a)

	test 1	test 2	test 3
h1	√	√	×
h2	√	√	×
h3	√	√	×
	√	√	×

(b)

	test 1	test 2	test 3
h1	√	×	×
h2	×	√	×
h3	×	×	√
	×	×	×

(c)

The more accurate and diverse the base learners, the better the ensemble.（好而不同）

Ensemble Learning

- However, the ‘’accuracy’’ and ‘’diversity’’ of base learner is conflicting.

- How to generate good base learners?

Outline

- ☐ Ensemble learning
- ☐ Ensemble method (How to generate base learners)
  - Bagging
  - Boosting
- ☐ Fusion Strategy
  - Averaging
  - Voting
  - Stacking
- ☐ Diversity

Ensemble methods

Parallel methods

- Bagging
- Random Forests
- … …

Sequencial methods

- Boosting
- … …

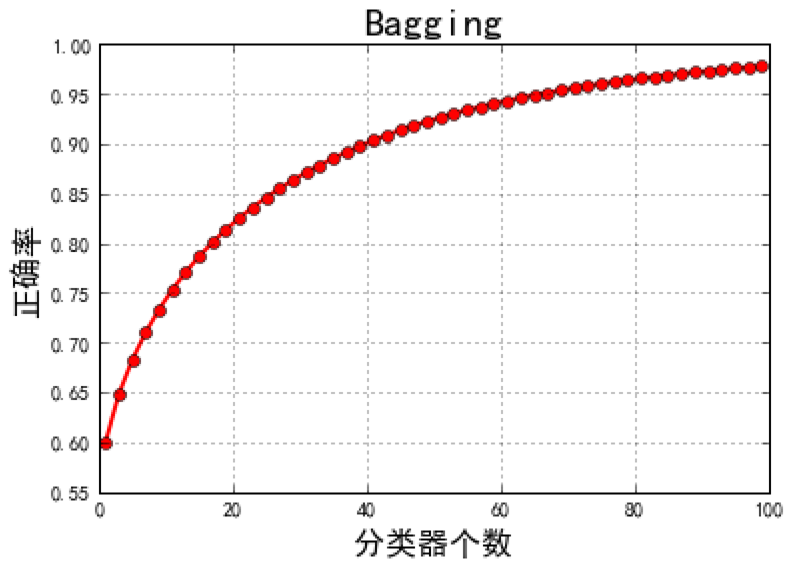
An example

- For a binary classify task, there are 5 independent classifiers with accuracy of 60% , we use voting strategy to combine all individual learners. What is the final classification accuracy?

$$P = C_5^3 \times 0.6^3 \times 0.4^2 + C_5^4 \times 0.6^4 \times 0.4^1 + C_5^5 \times 0.6^5 \times 0.4^0 = 0.68256$$

- If 99 classifiers?

```
9 :  0.73343232
19 :  0.813907978585
29 :  0.863787051336
39 :  0.897941368711
49 :  0.922424437652
59 :  0.940447995732
69 :  0.953949756505
79 :  0.964189692839
89 :  0.972027516007
99 :  0.97806955787
```



Bagging

- Bagging—Bootstrap aggregating
- Bagging is based on obtaining different training sets of equal size as the original one *L*, *by using a statistical* technique named bootstrap.
- The resulting training sets *L_i, i=1,…,n*, *usually contain small* changes with respect to *L*.
- $x^* = (x^*_1,…,x^*_n)$: random sample of size n drawn with replacement from the original sample $x = (x_1,…,x_n)$
- each sample in x can appear in x* zero times, once, twice, etc.

Bagging

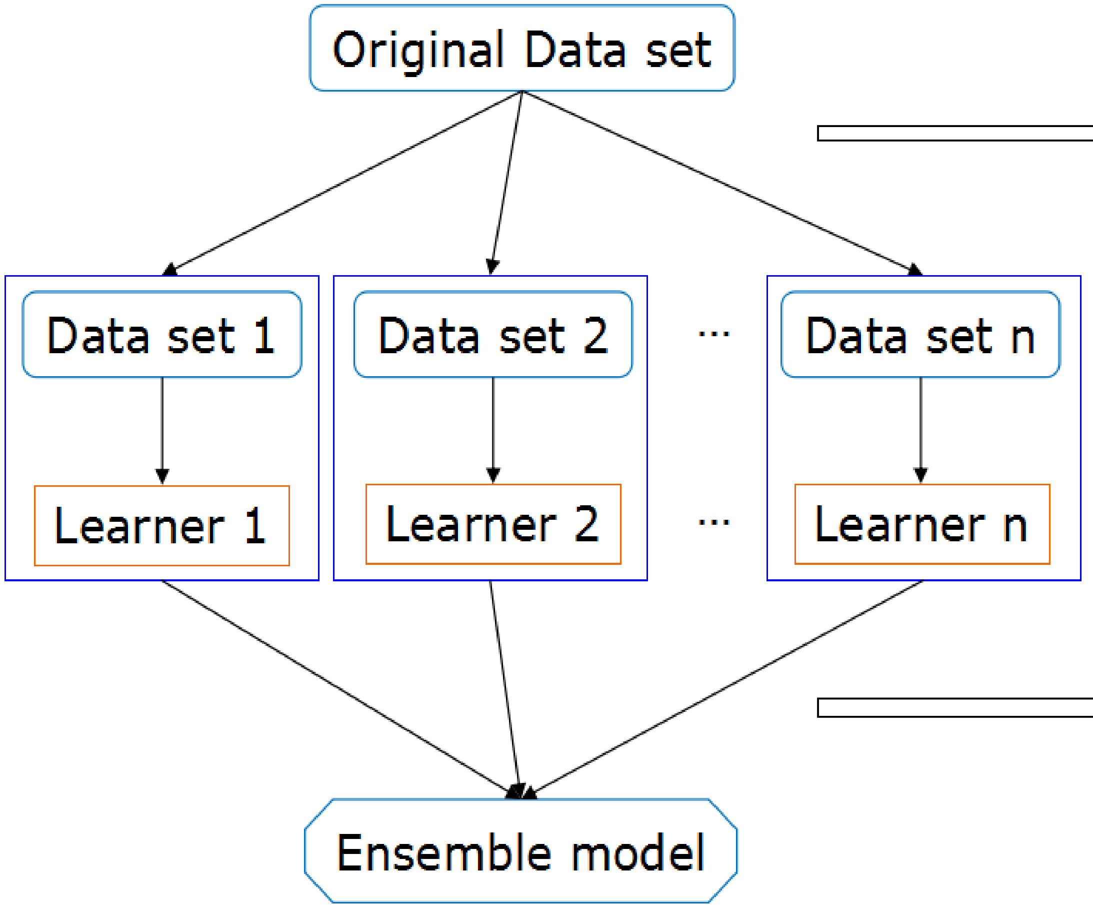
- unique instances : 63.2%

$$\lim_{n\to\infty}\left[1-\left(1-\frac{1}{n}\right)^{n}\right]=\lim_{n\to\infty}\left[1-\left(\left(1+\frac{1}{-n}\right)^{-n}\right)^{-1}\right]=1-e^{-1}\approx 0.632$$

- about 1/3 out-of-bag (OOB)
- Out-of-bag estimate

Only 63.2% samples of original training set are used in each training subset, so the 36.8% samples can be used as validation set.

Bagging



Bootstrap sampling: random sampling with replacement

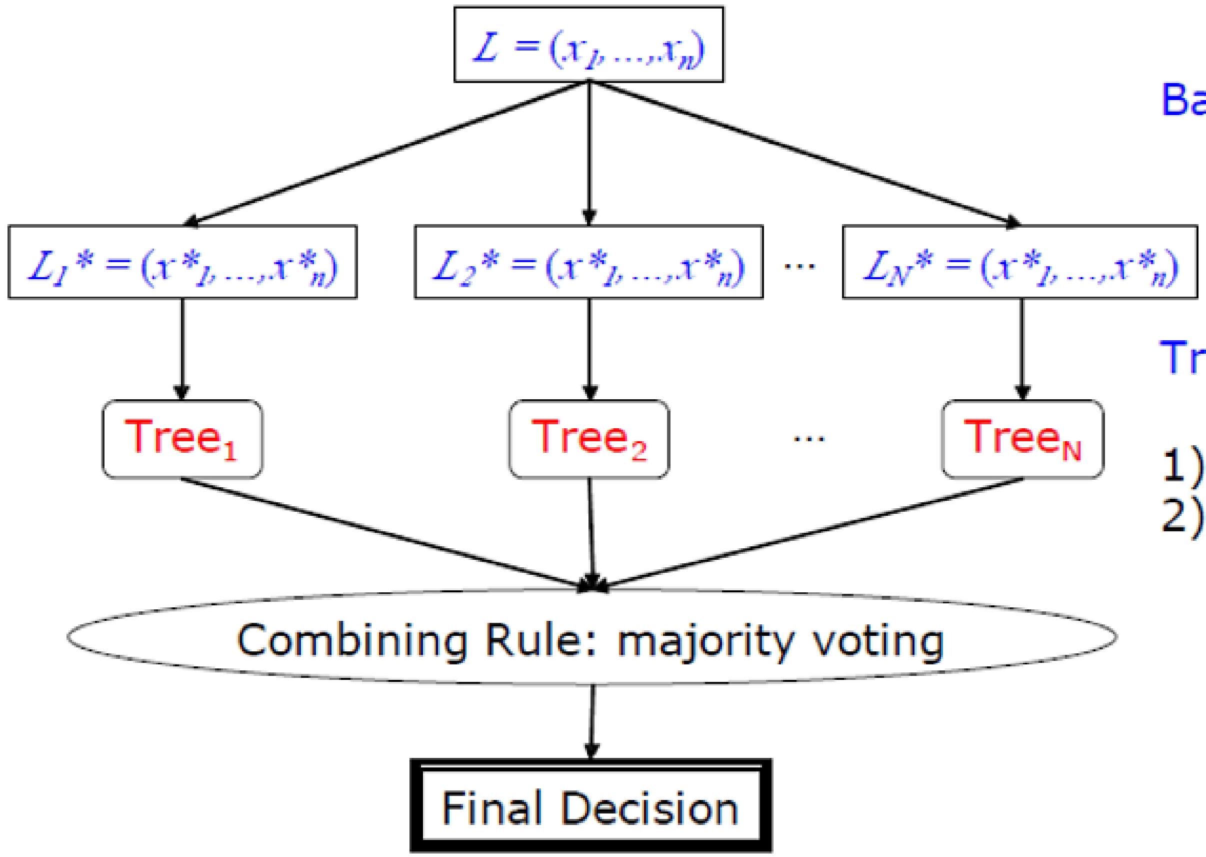
training individual learners

voting for classification
averaging for regression

Random Forest

- ❑ An extension of Bagging;
- ❑ Base learner: decision tree;
- ❑有放回构造训练集；随机选取m个特征构建决策树
- ❑ There are difference with traditional decision tree:
  - ❑ Introduce random property disturbance
  - ❑ Every tree is fully grown without pruning

Framework for Random Forest


$L = (x_1, \ldots, x_n)$
$L_1^* = (x^*_1, \ldots, x^*_n)$
$L_2^* = (x^*_1, \ldots, x^*_n)$
…
$L_N^* = (x^*_1, \ldots, x^*_n)$
Tree$_1$
Tree$_2$
…
Tree$_N$
Combining Rule: majority voting
Final Decision

Bagging process

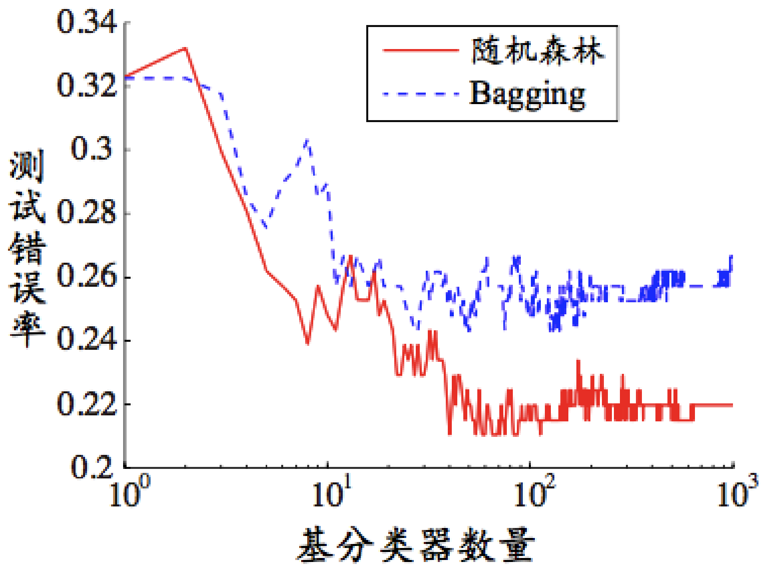
Tree growing process

1) no pruning;
2) m variables are used;

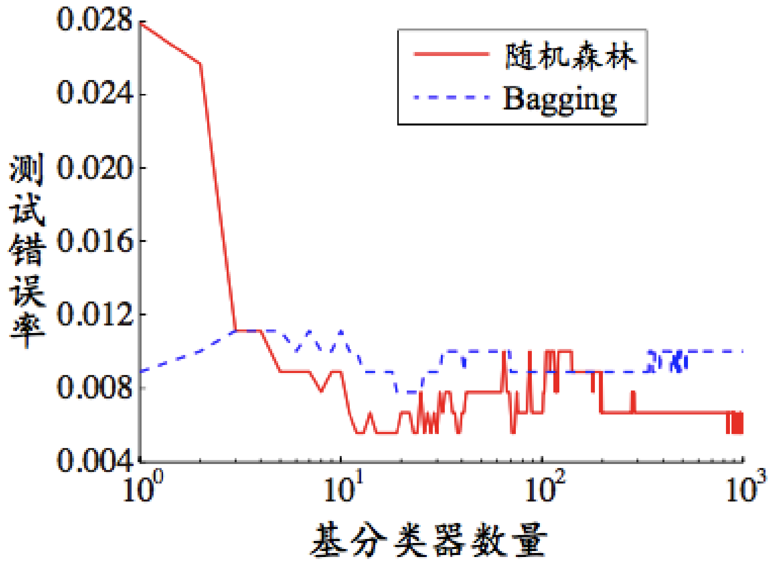
$$m = \log_2 M$$

M为总的属性个数

Random Forest实验



(a) glass 数据集



(b) auto-mpg 数据集

Bagging

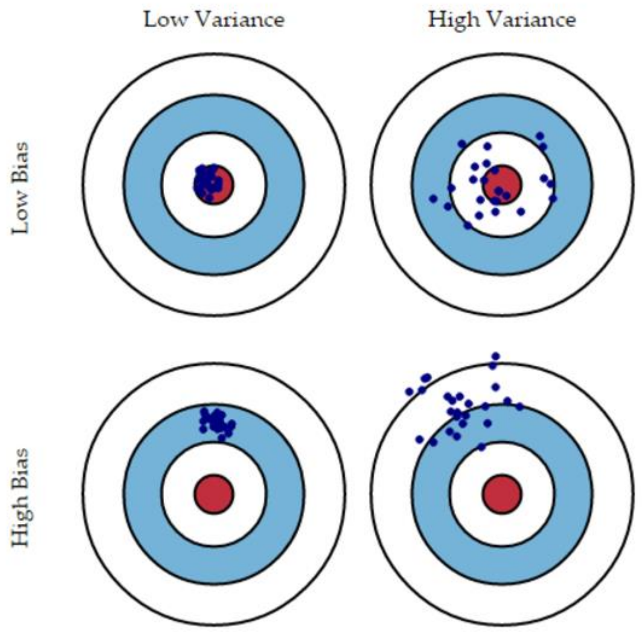
□Bagging ：

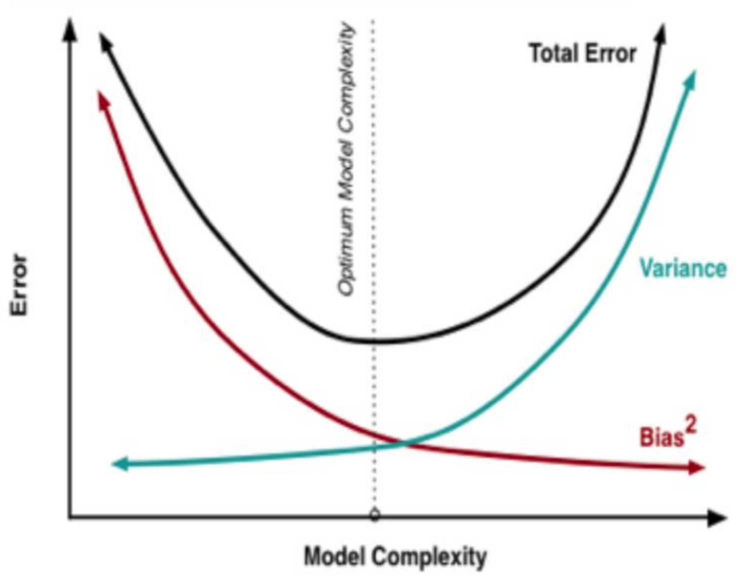
- Independent learners
- Parallel generated
- Bootstrap
- Decrease the variance

Bias & Variance

偏差度量了学习算法的期望预测与真实结果之间的偏离程度。即刻画了学习算法本身的拟合能力。

方差度量了同样大小的训练数据集的变动所导致的学习性能的变化，即刻画了数据扰动所造成的影响。





Ensemble methods

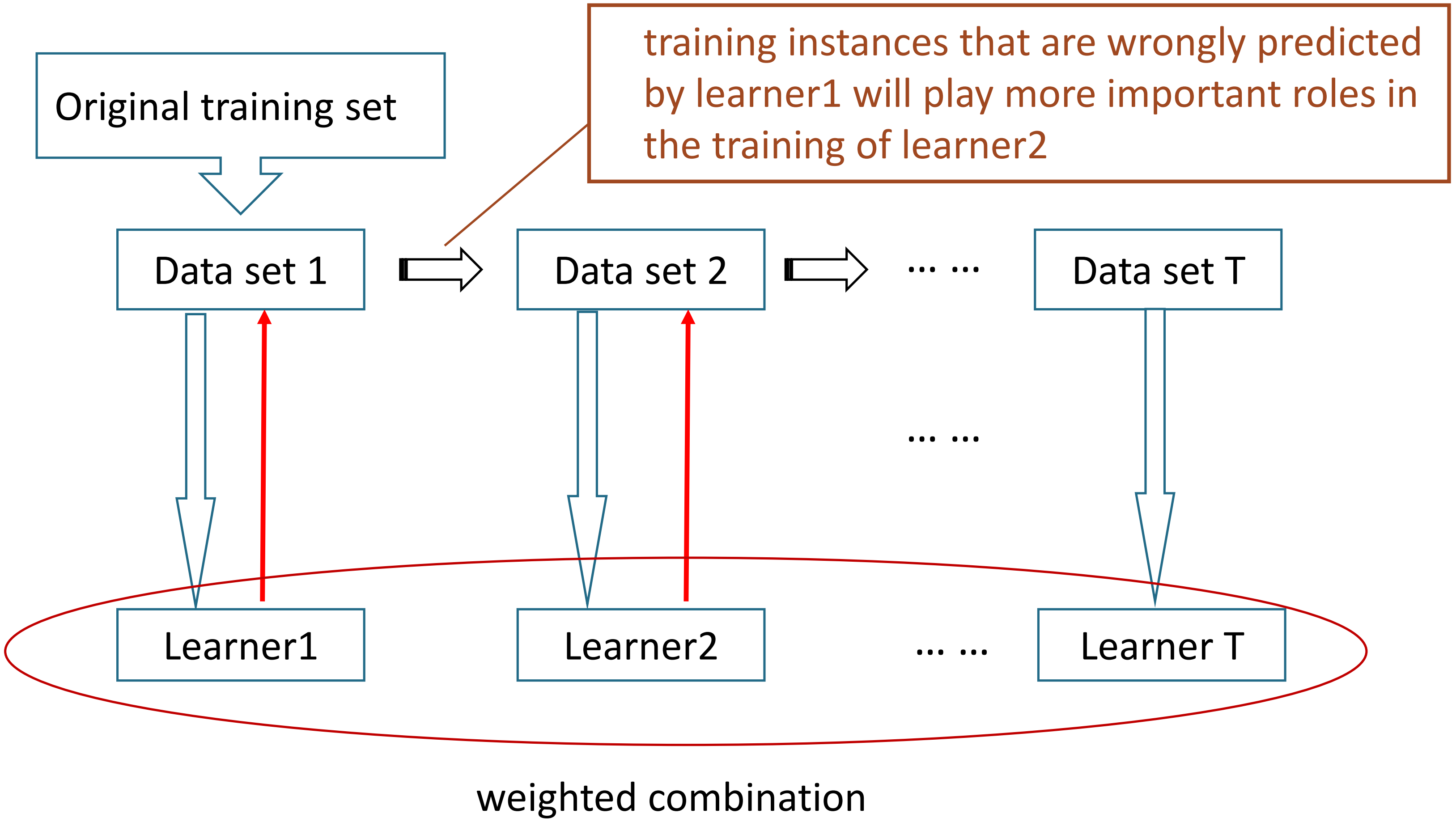
Parallel methods

- Bagging
- Random Forests
- … …

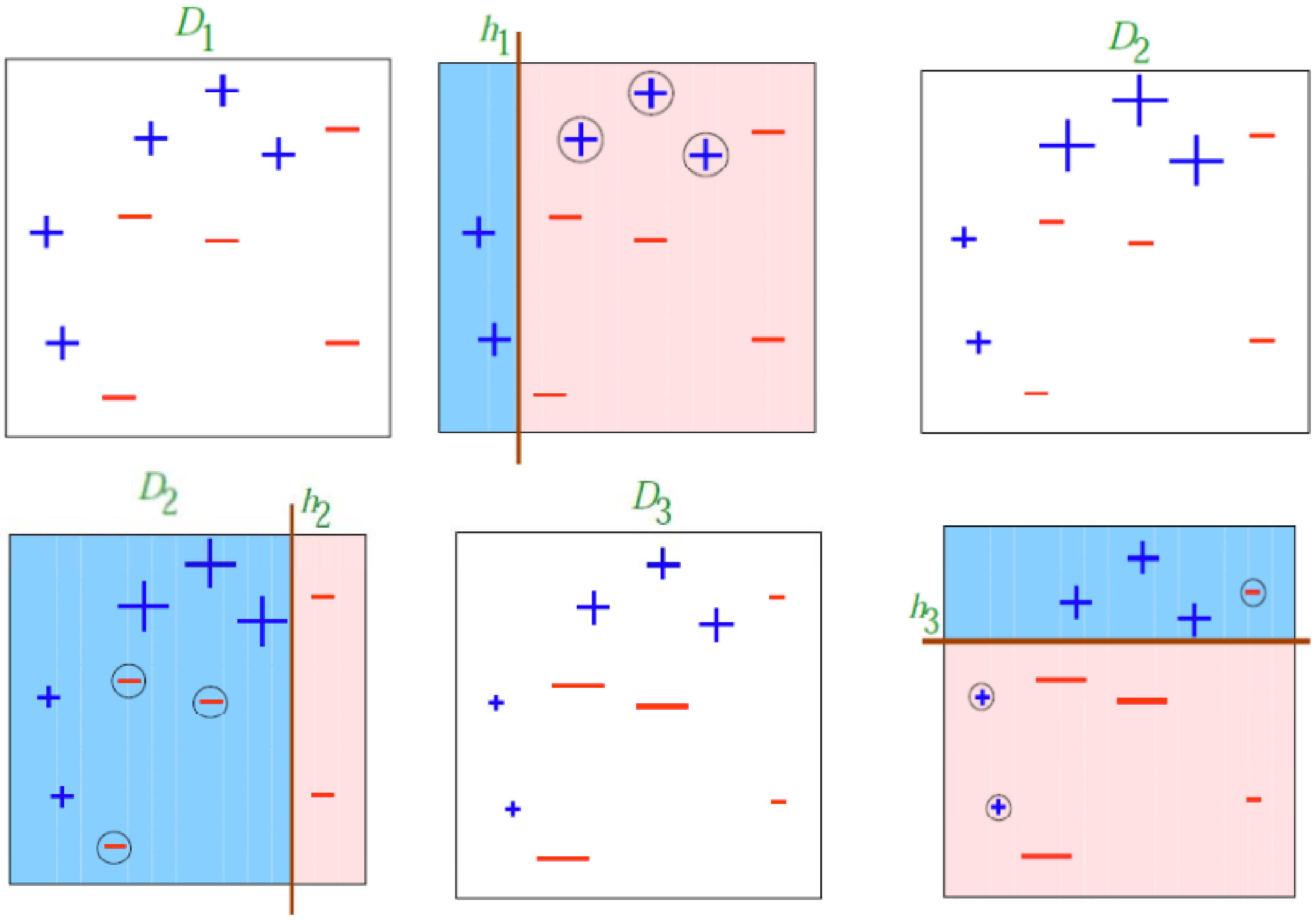
Sequencial methods: Boosting(提升)

- Adaboost
- Gradient boost: GBDT
- Xgboost
- … …

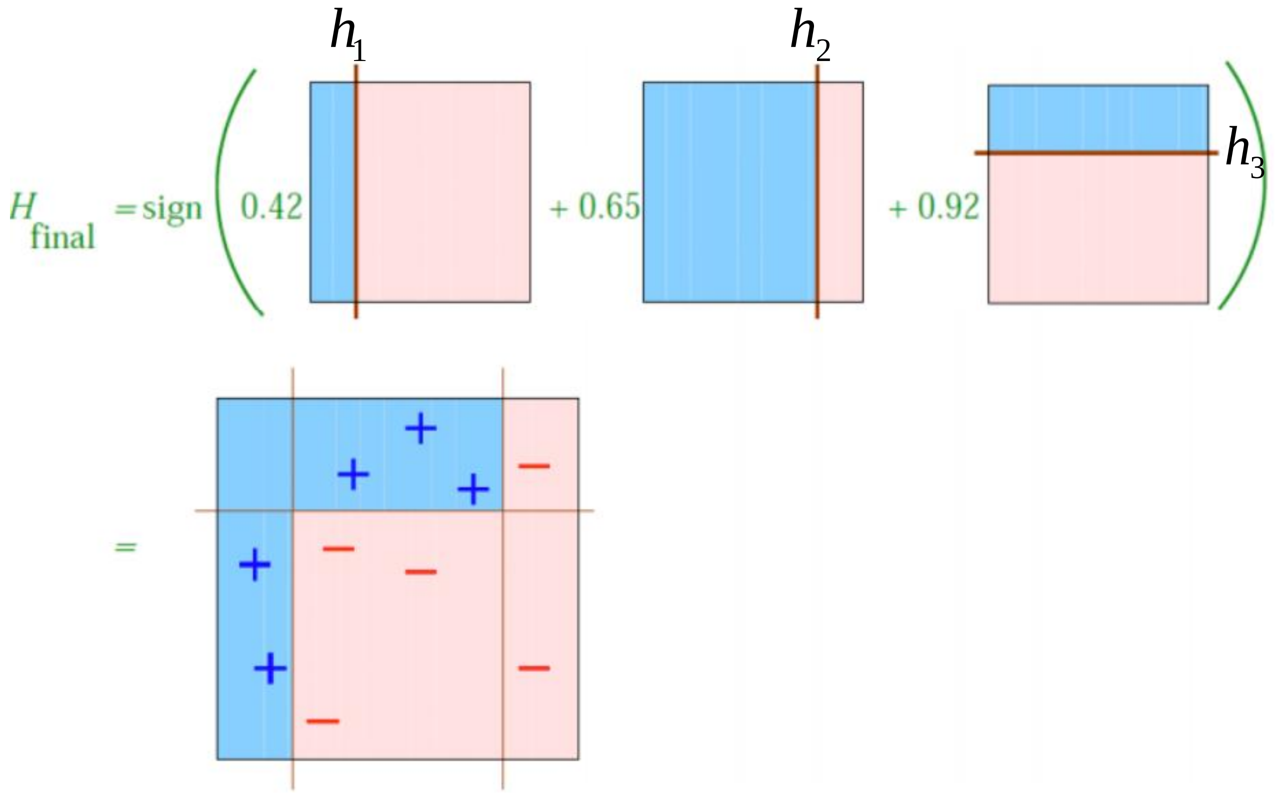
AdaBoost


Original training set
training instances that are wrongly predicted by learner1 will play more important roles in the training of learner2
Data set 1
Data set 2
… …
Data set T
… …
Learner1
Learner2
… …
Learner T
weighted combination

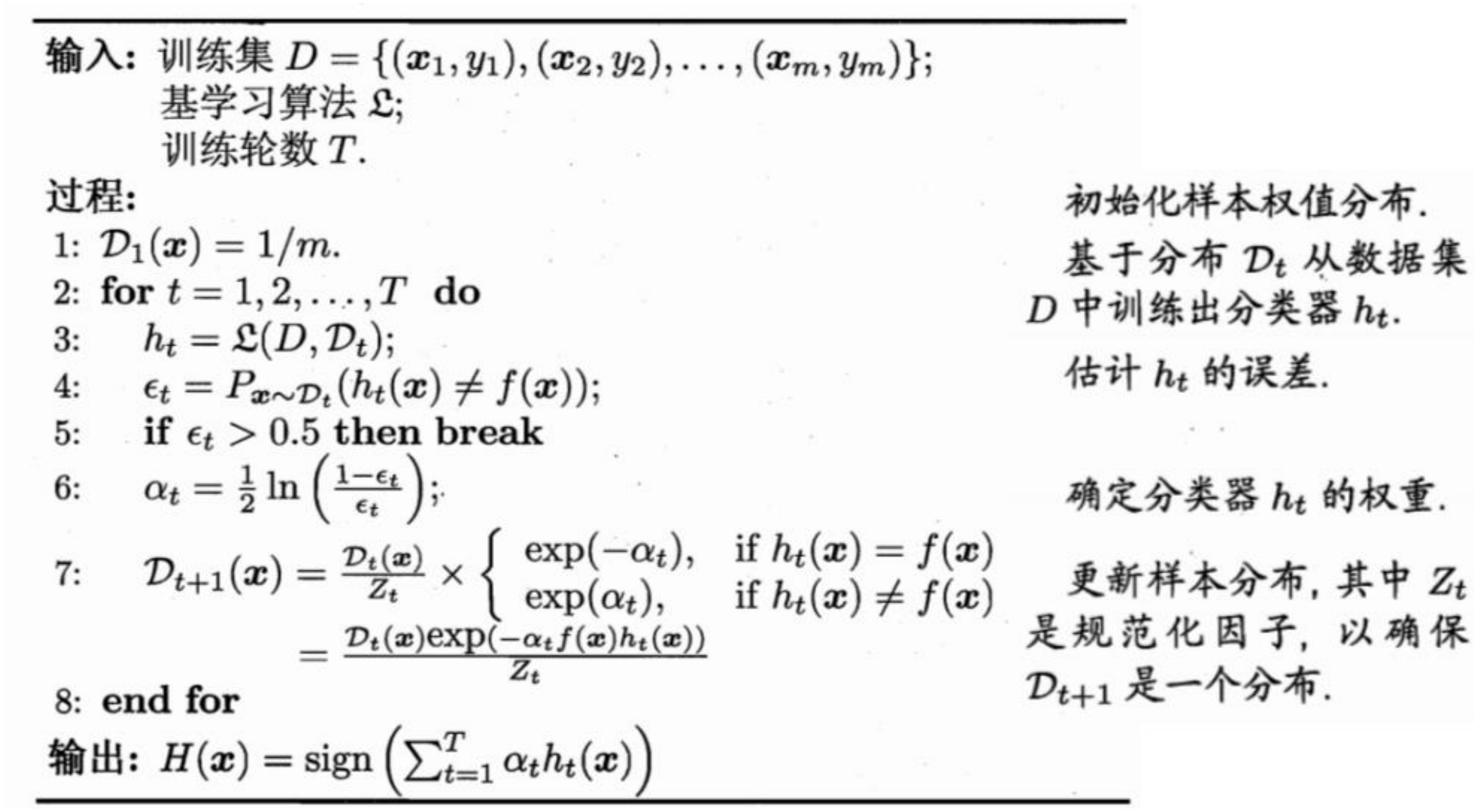
An Example



An Example


h_1
h_2
h_3
$H_{\text{final}} = \text{sign}\left(0.42 \quad + 0.65 \quad + 0.92 \quad\right)$
=

AdaBoost



```
输入: 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
      基学习算法 $\mathfrak{L}$;
      训练轮数 $T$.
过程:
1: $\mathcal{D}_1(\boldsymbol{x}) = 1/m$.
2: for $t = 1, 2, \ldots, T$ do
3:     $h_t = \mathfrak{L}(D, \mathcal{D}_t)$;
4:     $\epsilon_t = P_{\boldsymbol{x}\sim\mathcal{D}_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$;
5:     if $\epsilon_t > 0.5$ then break
6:     $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$;
7:     $\mathcal{D}_{t+1}(\boldsymbol{x}) = \frac{\mathcal{D}_t(\boldsymbol{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \exp(\alpha_t), & \text{if } h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}) \end{cases}$
       $= \frac{\mathcal{D}_t(\boldsymbol{x})\exp(-\alpha_t f(\boldsymbol{x}) h_t(\boldsymbol{x}))}{Z_t}$
8: end for
输出: $H(\boldsymbol{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})\right)$
```

初始化样本权值分布.

基于分布 $\mathcal{D}_t$ 从数据集 D 中训练出分类器 h_t.

估计 h_t 的误差.

确定分类器 h_t 的权重.

更新样本分布, 其中 Z_t 是规范化因子, 以确保 $\mathcal{D}_{t+1}$ 是一个分布.

AdaBoost

- Training set:

$$T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$$

- Initialize the weight distribution of all training samples:

$$D_1 = (w_{11}, w_{12}, \ldots, w_{1m}), \quad w_{1i} = \frac{1}{m}, i = 1,2,\cdots, m$$

AdaBoost

- Training from samples with weight distribution D_t , get the t_{th} base learner:

$$h_t(x) : \chi \rightarrow \{-1,+1\}$$

- Compute the classificaiton error rate of the $h_t(x)$ on the training set:

$$\varepsilon_t = P_{x\sim D_t}(h_t(x_i) \neq y_i) = \sum_{i=1}^{m} w_{ti} I(h_t(x_i) \neq y_i)$$

- Compute the weight of the classifier $h_t(x)$

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

AdaBoost

- Update the sample weight distribution on the training set:

$$D_{t+1} = (w_{t+1,1}, w_{t+1,2}, \ldots, w_{t+1,m})$$

$$w_{t+1,i} = \frac{w_{ti}}{Z_t} \times \exp(-\alpha_t \underbrace{y_i}_{\text{真实类别}} \overbrace{h_t}^{\text{预测类别}}(x_i)) \qquad \mathrm{i} = 1,2,\ldots,\mathrm{m}$$

- Z_t is a normalizized factor,which make the D_{t+1} be a probability distribuiton.

$$Z_t = \sum_{i=1}^{m} w_{ti} \exp(-\alpha_t y_i h_t(x_i))$$

AdaBoost

- Combine the learned base learners:

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

- the final classifier:

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

$$sign(x) = \begin{cases} 1 & , \mathrm{x} > 0 \\ 0 & , \mathrm{x} = 0 \\ -1 & , \mathrm{x} < 0 \end{cases}$$

An Example

- 给定下列训练样本，弱分类器由 $x < v$ 或 $x > v$ 产生，阈值 v 使得分类器在训练数据集上分类误差率最小。试用AdaBoost算法学习一个强分类器。

x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1

- 初始化数据权值分布

$$D_1 = (w_{11}, w_{12}, \ldots, w_{1m}), \quad w_{1i} = \frac{1}{m}, i = 1,2,\cdots,m$$

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

An Example

x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1
w	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

- 当训练轮数 t=1：
- 在权值分布为 D_1 的训练数据集上，阈值 V 取2.5时分类误差最小，所以基本分类器为

$$h_1(x) = \begin{cases} 1 & , x < 2.5 \\ -1 & , x > 2.5 \end{cases}$$

An Example

- $h_1(x)$ 在训练集上的误差率

$$\varepsilon_1 = P_{x\sim D_1}(h_t(x) \neq f(x)) = 0.3$$

- 计算 h_1 的权值

$$\alpha_1 = \frac{1}{2}\ln\left(\frac{1-\varepsilon_1}{\varepsilon_1}\right) = 0.4236$$

- 更新训练数据的权值分布

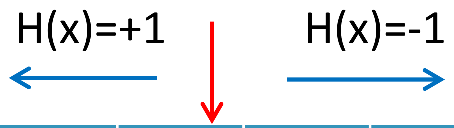
$$D_{t+1} = (w_{t+1,1}, w_{t+1,2}, \ldots, w_{t+1,m})$$

$$w_{t+1,i} = \frac{w_{ti}}{Z_t} \times \begin{cases} \exp(-\alpha_t), & if\ h_t(x) = y_i \\ \exp(\alpha_t), & if\ h_t(x) \neq y_i \end{cases}$$

An Example

$$D_2 = (0.0715, 0.0715, 0.0715, 0.0715, 0.0715, 0.0715, 0.1666, 0.1666, 0.1666, 0.0715)$$

- 此时，分类器 $H(x) = sign(0.4236 * h_1(x))$, $h_1(x) = \begin{cases} 1 & , x < 2.5 \\ -1 & , x > 2.5 \end{cases}$ 在训练集上有3个错分类点



x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1

An Example

x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1
w	0.0715	0.0715	0.0715	0.0715	0.0715	0.0715	0.1666	0.1666	0.1666	0.0715

- 在权值分布为 D_2 的训练数据上，阈值为8.5时分类错误率最低，因此基本分类器为

$$h_2(x)=\begin{cases}1 & ,x<8.5\\ -1, & x>8.5\end{cases}$$

An Example

- $h_2(x)$ 在训练数据上的误差率

$$\varepsilon_2 = P_{x\sim D_2}(h_2(x) \neq f(x)) = 0.2134\ (0.0715*3)$$

- 计算 $h_2(x)$ 的权值

$$\alpha_2 = \frac{1}{2}\ln\left(\frac{1-\varepsilon_2}{\varepsilon_2}\right) = 0.6496$$

- 更新训练数据的权值分布

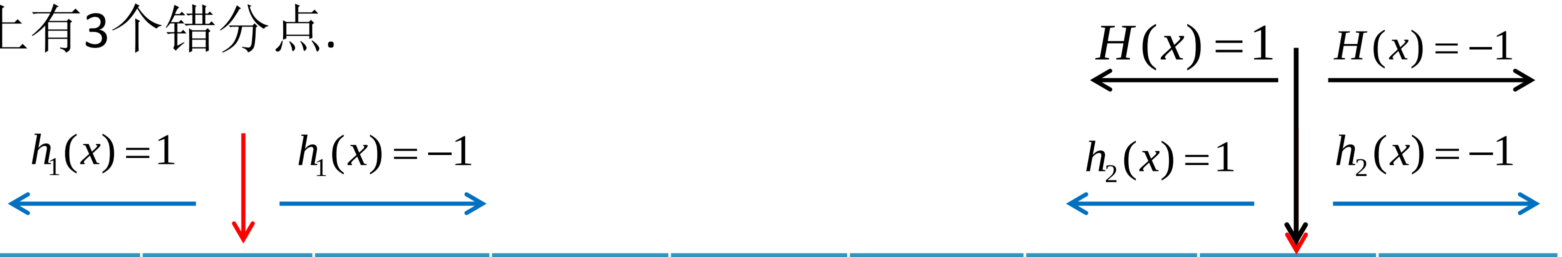
$$D_{t+1} = (w_{t+1,1}, w_{t+1,2}, \ldots, w_{t+1,m})$$

$$w_{t+1,i} = \frac{w_{ti}}{Z_t} \times \begin{cases} \exp(-\alpha_t), & if\ h_t(x) = y_i \\ \exp(\alpha_t), & if\ h_t(x) \neq y_i \end{cases}$$

An Example

$$D_3 = (0.0455, 0.0455, 0.0455, 0.1667, 0.01667, 0.1060, 0.1060, 0.1060, 0.0455)$$

- 此时，分类器 $H(x) = sign\,(0.4236 * h_1(x) + 0.6496 * h_2(x))$ 在训练集上有3个错分点.



x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1
w	0.0715	0.0715	0.0715	0.0715	0.0715	0.0715	0.1666	0.1666	0.1666	0.0715

An Example

x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1
w	0.0455	0.0455	0.0455	0.1667	0.1667	0.1667	0.1060	0.1060	0.1060	0.0455

- 在权值分布为D_3 的训练数据上，阈值取5.5分类误差率最低，因此基本分类器为：

$$h_3(x)=\begin{cases}-1, x<5.5\\ 1\ \ , x>5.5\end{cases}$$

An Example

- $h_3(x)$ 在训练数据上的误差率

$$\varepsilon_3 = P_{x\sim D_3}(h_3(x) \neq f(x)) = 0.1820\ (0.0455*4)$$

- 计算 $h_3(x)$ 的权值

$$\alpha_3 = \frac{1}{2}\ln\left(\frac{1-\varepsilon_3}{\varepsilon_3}\right) = 0.7514$$

An Example

- 更新训练数据的权值分布：

$$D_{t+1} = (w_{t+1,1}, w_{t+1,2}, \ldots, w_{t+1,m})$$

$$w_{t+1,i} = \frac{w_{ti}}{Z_t} \times \begin{cases} \exp(-\alpha_t), & if\ h_t(x) = y_i \\ \exp(\alpha_t), & if\ h_t(x) \neq y_i \end{cases}$$

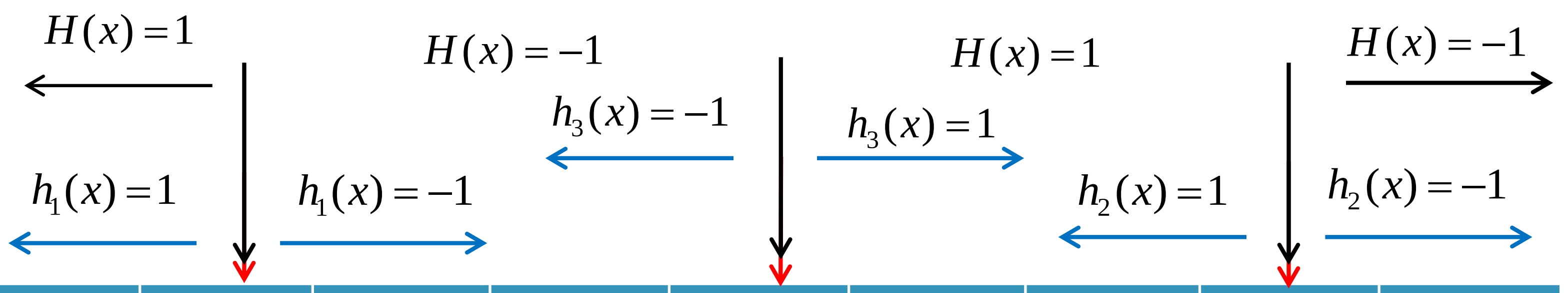
$$D_4 = (0.125, 0.125, 0.125, 0.102, 0.102, 0.102, 0.065, 0.065, 0.065, 0.125)$$

$$H(x) = sign(0.4236 * h_1(x) + 0.6496 * h_2(x) + 0.7514 * h_3(x))$$

- 在训练集上有0个误分类点

An Example

$$H(x) = sign\,(0.4236 * h_1(x) + 0.6496 * h_2(x) + 0.7514 * h_3(x))$$



x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1
w	0.0455	0.0455	0.0455	0.1667	0.1667	0.1667	0.1060	0.1060	0.1060	0.0455

Bagging VS Boosting

☐Bagging ：

- Independence between base learners
- parallel generated
- Bootstrap
- Decrease the variance

☐AdaBoost：

- Dependence between base learners
- Sequence generated
- Adjust the weight distribution with samples
- Decrease the bias（overfitting）

Outline

- ☐ Ensemble learning
- ☐ Ensemble method
  - Boosting
  - Bagging
- ☐ Fusion strategy (how to combine the base learners)
  - Averaging
  - Voting
  - Stacking
- ☐ Diversity

Fusion Strategy—Averaging

- simple averaging

$$H(\boldsymbol{x}) = \frac{1}{T}\sum_{i=1}^{T} h_i(\boldsymbol{x})$$

- weighted averaging

$$H(\boldsymbol{x}) = \sum_{i=1}^{T} w_i h_i(\boldsymbol{x}) \qquad w_i \geqslant 0,\ \sum_{i=1}^{T} w_i = 1$$

一般而言，在个体学习器性能相差较大时宜使用加权平均法，个体学习器性能相近时宜使用简单平均法。

Fusion Strategy—Voting

学习器 h_i 将从类别标记集合 $\{c_1, c_2, \ldots, c_N\}$ 中预测出一个标记, 我们将 h_i 在样本 $\boldsymbol{x}$ 上的预测输出表示为一个 N 维向量 $(h_i^1(\boldsymbol{x}); h_i^2(\boldsymbol{x}); \ldots; h_i^N(\boldsymbol{x}))$, 其中 $h_i^j(\boldsymbol{x})$ 是 h_i 在类别标记 c_j 上的输出.

- majority voting （绝对多数投票法）

$$H(\boldsymbol{x}) = \begin{cases} c_j & \text{if } \sum_{i=1}^{T} h_i^j(\boldsymbol{x}) > \frac{1}{2} \sum_{k=1}^{l} \sum_{i=1}^{T} h_i^k(\boldsymbol{x}) \\ \text{rejection} & \text{otherwise} . \end{cases}$$

- plurality voting（相对多数投票法）

$$H(\boldsymbol{x}) = c_{\underset{j}{\arg\max} \sum_{i=1}^{T} h_i^j(\boldsymbol{x})}$$

- weighted voting（加权投票法）

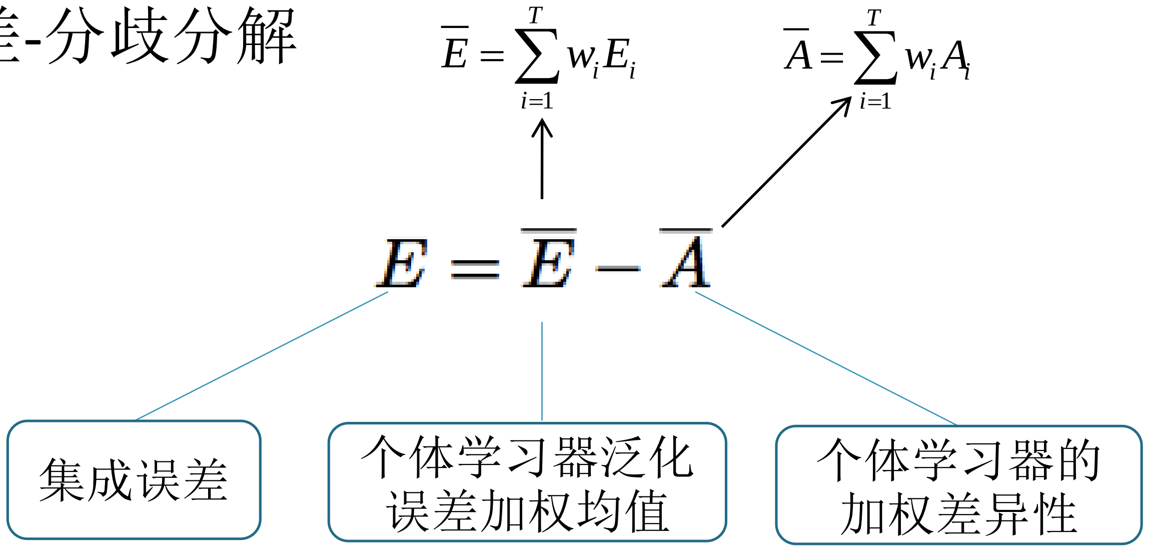
$$H(\boldsymbol{x}) = c_{\underset{j}{\arg\max} \sum_{i=1}^{T} w_i h_i^j(\boldsymbol{x})}$$

Outline

- ☐ Ensemble learning
- ☐ Ensemble method
  - Boosting
  - Bagging
- ☐ Blending
  - Averaging
  - Voting
  - Stacking
- ☐ Diversity

Diversity

- 误差-分歧分解


$\overline{E} = \sum_{i=1}^{T} w_i E_i$
$\overline{A} = \sum_{i=1}^{T} w_i A_i$
$E = \overline{E} - \overline{A}$
集成误差
个体学习器泛化误差加权均值
个体学习器的加权差异性

个体学习器的准确率越高，多样性越大，则集成越好。

Diversity Measure

顾名思义, 多样性度量(diversity measure)是用于度量集成中个体分类器的多样性, 即估算个体学习器的多样化程度. 典型做法是考虑个体分类器的两两相似/不相似性.

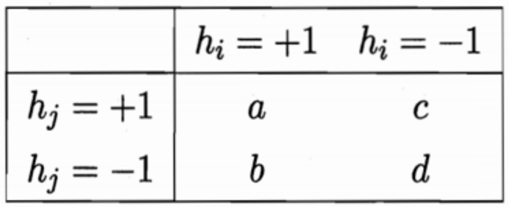
给定数据集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$, 对二分类任务, $y_i \in \{-1, +1\}$, 分类器 h_i 与 h_j 的预测结果列联表(contingency table)为

	$h_i = +1$	$h_i = -1$
$h_j = +1$	a	c
$h_j = -1$	b	d

其中, a 表示 h_i 与 h_j 均预测为正类的样本数目; b、c、d 含义由此类推; $a + b + c + d = m$. 基于这个列联表, 下面给出一些常见的多样性度量.

Diversity Measure

- Disagreement Measure（不合度量）



	$h_i = +1$	$h_i = -1$
$h_j = +1$	a	c
$h_j = -1$	b	d

$$dis_{ij} = \frac{b+c}{m}$$

dis_{ij} 的值域为 $[0,1]$. 值越大则多样性越大.

- Correlation Coefficient（相关系数）

$$\rho_{ij} = \frac{ad-bc}{\sqrt{(a+b)(a+c)(c+d)(b+d)}}$$

ρ_{ij} 的值域为 $[-1,1]$. 若 h_i 与 h_j 无关, 则值为 0; 若 h_i 与 h_j 正相关则值为正, 否则为负.

p 的绝对值越大，两个分类器的相关性越大，反之，相关性越小。

Diversity Measure

	$h_i = +1$	$h_i = -1$
$h_j = +1$	a	c
$h_j = -1$	b	d

- Q-Statistic （Q-统计量）

$$Q_{ij} = \frac{ad - bc}{ad + bc}$$

$$|Q_{ij}| \leq |\rho_{ij}|$$

- Kappa-Statistic（K-统计量）

$$\kappa = \frac{p_1 - p_2}{1 - p_2}$$

$$p_1 = \frac{a+d}{m},$$

$$p_2 = \frac{(a+b)(a+c)+(c+d)(b+d)}{m^2}$$

p1:两个分类器取得一致的概率

p2:两个分类器偶然达成一致的概率

K 越大，相关性越大，反之，相关性越小。

Diversity Enhanced

- 在集成学习中需有效的生成多样性大的个体学习器，一般思路是在学习过程中引入随机性。
  - 数据样本扰动
  - 输入属性扰动
  - 输出表示扰动
  - 算法参数扰动

Diversity Enhanced

- 数据样本扰动

给定初始数据集，可从中产生出不同的数据子集，再利用不同的数据子集训练出不同的个体学习器。数据样本扰动通常是基于采样法：Boosting 、Bagging

对于不稳定基学习器（决策树、神经网络…）很有效。

- 输入属性扰动

对于稳定基学习器（LDA、SVM、KNN…），从初始属性集中抽取出若干个属性子集，基于每个属性子集训练一个基学习器。

对包含大量冗余属性的数据，在子空间中训练个体学习器不仅能产生多样性大的个体，还会因属性数的减少而节省时间开销。

Diversity Enhanced

- 输出表示扰动

　将原任务拆解为多个可同时求解的子任务或者是将分类输出转化为回归输出后构建个体学习器。

- 算法参数扰动

　基学习算法一般都有参数需要设置，例如神经网络的神经元数，初始连接权值等，通过随机设置不同的参数。往往可产生差别较大的个体学习器。

More about ensemble methods

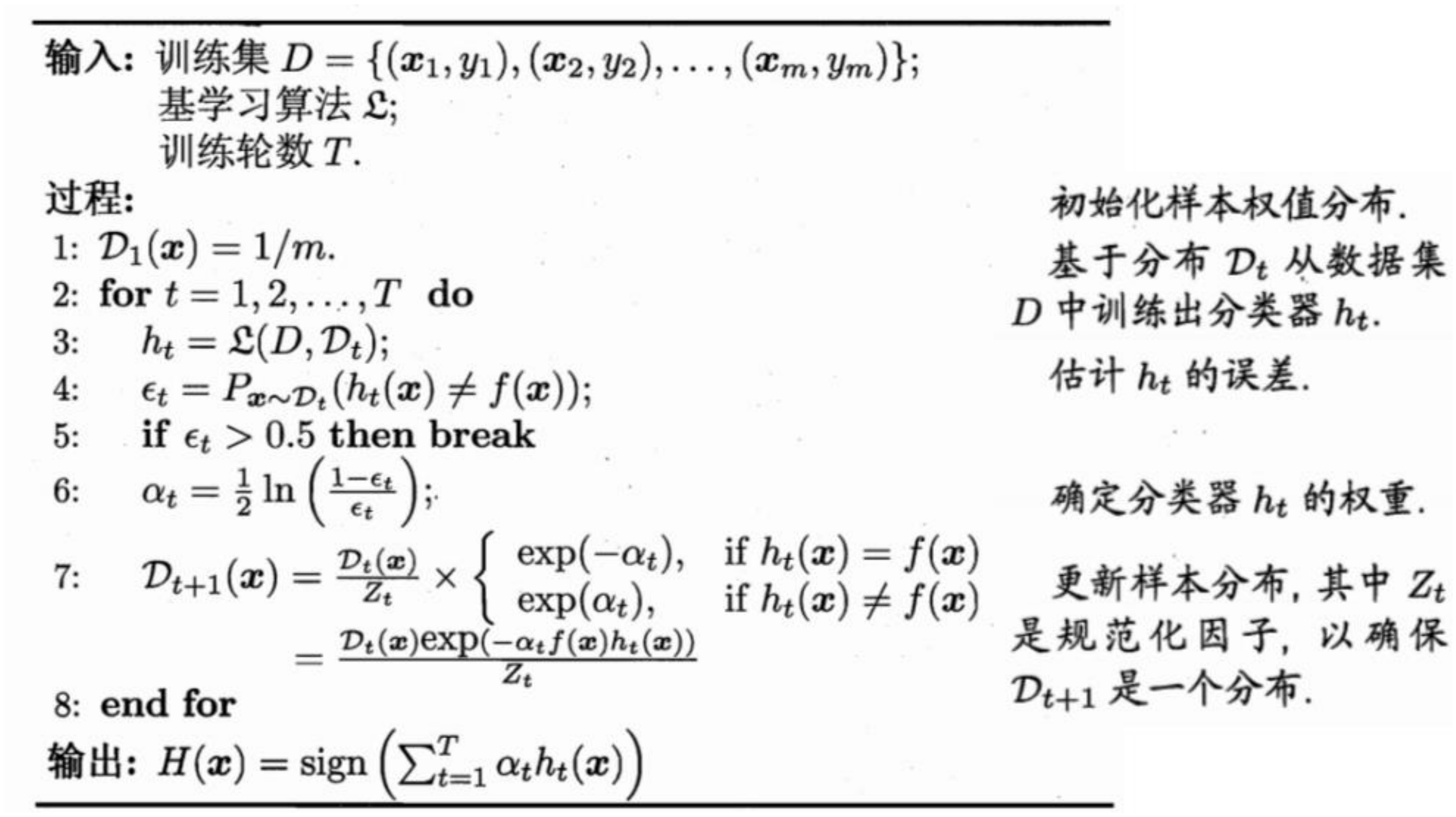
- 周志华《机器学习》
- Z.-H. Zhou.Ensemble Methods:Foundations and Algorithms,Boca Raton, FL: Chapman &Hall/CRC, Jun. 2012.(ISBN 978-1-439-830031)
- 李航 《统计学习方法》
- 机器学习公开课程　台湾大学 林轩田
- 机器学习课程 coursera　吴恩达

  https://www.coursera.org/learn/machine-learning#

Homework

- 试分析随机森林为什么比决策树bagging集成的训练速度更快？
- 了解GBDT、Xgboost 原理

附：AdaBoost推导



```
输入: 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
      基学习算法 $\mathfrak{L}$;
      训练轮数 $T$.
过程:
1: $\mathcal{D}_1(\boldsymbol{x}) = 1/m$.
2: for $t = 1, 2, \ldots, T$ do
3:     $h_t = \mathfrak{L}(D, \mathcal{D}_t)$;
4:     $\epsilon_t = P_{\boldsymbol{x} \sim \mathcal{D}_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$;
5:     if $\epsilon_t > 0.5$ then break
6:     $\alpha_t = \frac{1}{2} \ln \left( \frac{1-\epsilon_t}{\epsilon_t} \right)$;
7:     $\mathcal{D}_{t+1}(\boldsymbol{x}) = \frac{\mathcal{D}_t(\boldsymbol{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\boldsymbol{x}) = f(\boldsymbol{x}) \\ \exp(\alpha_t), & \text{if } h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}) \end{cases}$
       $= \frac{\mathcal{D}_t(\boldsymbol{x}) \exp(-\alpha_t f(\boldsymbol{x}) h_t(\boldsymbol{x}))}{Z_t}$
8: end for
输出: $H(\boldsymbol{x}) = \operatorname{sign}\left( \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \right)$
```

初始化样本权值分布.

基于分布 $\mathcal{D}_t$ 从数据集 D 中训练出分类器 h_t.

估计 h_t 的误差.

确定分类器 h_t 的权重.

更新样本分布, 其中 Z_t 是规范化因子, 以确保 $\mathcal{D}_{t+1}$ 是一个分布.

前向分步算法

- 优化目标：

  最小化指数损失函数

$$f(x) \in \{-1,+1\}$$

$$l_{\exp}(H \mid D) = \boxed{E_{x \sim D}[e^{-\boxed{f(x)}\boxed{H(x)}}]}$$

分布D中的所有样本x的期望损失

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

Loss Function

- 集成的学习器$H(x)$能使指数损失函数最小化，即：

$$\frac{\partial \ell_{\exp}(H \mid \mathcal{D})}{\partial H(\boldsymbol{x})} = E_{x\sim D}\left(-f(x)\exp(-f(x)H(x))\right)$$

$$\Longrightarrow \quad \frac{\partial \ell_{\exp}(H \mid \mathcal{D})}{\partial H(\boldsymbol{x})} = -e^{-H(\boldsymbol{x})} P(f(\boldsymbol{x})=1 \mid \boldsymbol{x}) + e^{H(\boldsymbol{x})} P(f(\boldsymbol{x})=-1 \mid \boldsymbol{x}) = 0$$

$$\Longrightarrow \quad H(\boldsymbol{x}) = \frac{1}{2} \ln \frac{P(f(x)=1 \mid \boldsymbol{x})}{P(f(x)=-1 \mid \boldsymbol{x})}$$

Loss Function

$$\begin{aligned}\operatorname{sign}(H(\boldsymbol{x})) &= \operatorname{sign}\left(\frac{1}{2}\ln\frac{P(f(x)=1\mid\boldsymbol{x})}{P(f(x)=-1\mid\boldsymbol{x})}\right)\\ &= \begin{cases}1, & P(f(x)=1\mid\boldsymbol{x}) > P(f(x)=-1\mid\boldsymbol{x})\\ -1, & P(f(x)=1\mid\boldsymbol{x}) < P(f(x)=-1\mid\boldsymbol{x})\end{cases}\\ &= \underset{y\in\{-1,1\}}{\arg\max}\, P(f(x)=y\mid\boldsymbol{x}),\end{aligned}$$

$sign(H(x))$达到了贝叶斯最优错误率，说明指数损失函数是分类任务原来0/1损失函数的替代函数。

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

Bayes optimal classifier

- 最小化分类错误率的贝叶斯最优分类器：

$$h(x) = \arg\max_{c \in y} P(c \mid x) \qquad (7.6)$$

对每个样本x, 选择能使后验概率 $P(c \mid x)$ 最大的类别标记。

- 其中，分类错误率对应0/1损失函数：

$$L(Y, f(x)) = \begin{cases} 1, & \mathrm{Y} \neq \mathrm{f(x)} \\ 0, & \mathrm{Y} = \mathrm{f(x)} \end{cases}$$

前向分步算法

- 基学习器线性组合组成最终的分类器：

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

- 假设经过t-1轮迭代前向分步算法已经得到 $H_{t-1}(x)$ ：

$$\begin{aligned}H_{t-1}(x)&=H_{t-2}(x)+\alpha_{t-1}h_{t-1}(x)\\&=\alpha_1 h_1(x)+\alpha_2 h_2(x)+\cdots+\alpha_{t-1}h_{t-1}(x)\end{aligned}$$

- 在第 t 轮得到 $\alpha_t, h_t(x)$和$H_t(x)$ ， 即

$$H_t(x)=H_{t-1}(x)+\alpha_t h_t(x)$$

前向分步算法

$$L(y,H(x))=\sum_{i=1}^{m}\exp(-y_i(H(x_i)))$$

$$=\sum_{i=1}^{m}\exp(-y_i(H_{t-1}(x_i)+\alpha_t h_t(x_i))) \qquad \text{令}\overline{w}_{ti}=\exp(-y_i(H_{t-1}(x_i)))$$

$$=\sum_{i=1}^{m}\overline{w}_{ti}\exp(-y_i\alpha_t h_t(x_i))=\sum_{y_i=h_t(x_i)}\overline{w}_{ti}e^{-\alpha_t}+\sum_{y_i\neq h_t(x_i)}\overline{w}_{ti}e^{\alpha_t}$$

$$=\sum_{i=1}^{m}e^{-\alpha_t}\overline{w}_{ti}I(y_i=h_t(x_i))+\sum_{i=1}^{m}e^{\alpha_t}\overline{w}_{ti}I(y_i\neq h_t(x_i))$$

$$=(e^{\alpha_t}-e^{-\alpha_t})\sum_{i=1}^{m}\overline{w}_{ti}I(y_i\neq h_t(x_i))+e^{-\alpha_t}\sum_{i=1}^{m}\overline{w}_{ti} \qquad (*)$$

前向分步算法

- 令 $\frac{\partial L}{\partial \alpha_t}=0$ 得到：

$$(e^{\alpha_t}+e^{-\alpha_t})\sum_{i=1}^{m}\overline{w}_{ti}I(y_i\neq h_t(x_i))=e^{-\alpha_t}\sum_{i=1}^{m}\overline{w}_{ti}$$

$$\Rightarrow (e^{\alpha_t}+e^{-\alpha_t})\boxed{\frac{\sum_{i=1}^{m}\overline{w}_{ti}I(y_i\neq h_t(x_i))}{\sum_{i=1}^{m}\overline{w}_{ti}}}=e^{-\alpha_t}$$

$$\rightarrow \sum_{i=1}^{m}w_{ti}I(y_i\neq h_t(x_i))=\varepsilon_t$$

$$e^{\alpha_t}\varepsilon_t+e^{-\alpha_t}\varepsilon_t=e^{-\alpha_t}$$

$$\Rightarrow e^{\alpha_t}\varepsilon_t=e^{-\alpha_t}(1-\varepsilon_t)$$

$$\Rightarrow e^{\alpha_t}=e^{-\alpha_t}\frac{(1-\varepsilon_t)}{\varepsilon_t} \Longrightarrow \alpha_t=\frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

前向分步算法

令$\overline{w}_{ti} = \exp(-y_i(H_{t-1}(x_i)))$

$$\overline{w}_{t+1,i} = \exp(-y_i H_t(x_i))$$
$$= \exp(-y_i(H_{t-1}(x_i) + \alpha_t h_t(x_i)))$$
$$= \overline{w}_{ti} \exp(-y_i \alpha_t h_t(x_i))$$

此时， $\overline{w}_{t+1,i} = \overline{w}_{ti} \times \exp\,(-\alpha_t y_i h_t(x_i))$

$$w_{t+1,i} = \frac{w_{ti}}{Z_t} \times \exp(-\alpha_t y_i h_t(x_i)) \qquad \mathrm{i = 1,2,\ldots,m}$$

Thanks !