

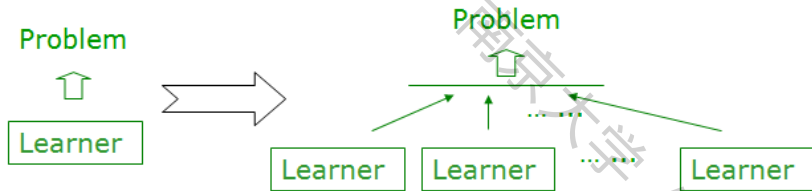
八、集成学习

主讲教师：周志华

集成学习

Ensemble Learning (集成学习):

Using multiple learners to solve the problem



Demonstrated great performance in real practice

- ❑ KDDCup'07: 1st place for "... Decision Forests and ..."
- ❑ KDDCup'08: 1st place of Challenge1 for a method using Bagging; 1st place of Challenge2 for "... Using an Ensemble Method "
- ❑ KDDCup'09: 1st place of Fast Track for "Ensemble ..."; 2nd place of Fast Track for "... bagging ... boosting tree models ..."; 1st place of Slow Track for "Boosting ..."; 2nd place of Slow Track for "Stochastic Gradient Boosting"
- ❑ KDDCup'10: 1st place for "... Classifier ensembling"; 2nd place for "... Gradient Boosting machines ..."
- ❑ 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"
- ❑ 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..."
- ❑ KDDCup'15: 1st place for "Three-Stage Ensemble and Feature Engineering for MOOC Dropout Prediction"
- ❑ KDDCup'16: 1st place for "Gradient Boosting Decision Tree"; 2nd place for "Ensemble of Different Models for Final Prediction"
- ❑ KDDCup'17: 1st and 2nd place of Task 1 for "XGBoost"; 1st place of Task 2 for "XGBoost", 2nd place of Task 2 for "Weighted Average of Multiple Models"
- ❑ KDDCup'18: 1st place for "Gradient Boosting"; 2nd place for "Two-stage stacking"; 3rd place for "Weighted Average of Multiple Models"

During the past decade, almost all winners of KDDCup, Netflix competition, Kaggle competitions, etc., utilized ensemble techniques in their solutions

To win? Ensemble !

如何得到好的集成？

测试例1 测试例2 测试例3				测试例1 测试例2 测试例3				测试例1 测试例2 测试例3			
h_1	✓	✓	×	h_1	✓	✓	×	h_1	✓	×	×
h_2	×	✓	✓	h_2	✓	✓	×	h_2	×	✓	×
h_3	✓	×	✓	h_3	✓	✓	×	h_3	×	×	✓
集成	✓	✓	✓	集成	✓	✓	×	集成	×	×	×
(a) 集成提升性能				(b) 集成不起作用				(c) 集成起负作用			

令个体学习器“好而不同”

“多样性” (diversity) 是关键

误差-分歧分解 (error-ambiguity decomposition):

$$E = \bar{E} - \bar{A}$$

Diagram illustrating the error-ambiguity decomposition:

- E (Ensemble error) is represented by a black box.
- \bar{E} (Ave. error of individuals) is represented by a green box.
- \bar{A} (Ave. “ambiguity” of individuals) is represented by a red box.

Arrows indicate the mapping from the boxes to their respective labels:

- Black arrow from E to *Ensemble error*.
- Green arrow from \bar{E} to *Ave. error of individuals*.
- Red arrow from \bar{A} to *Ave. “ambiguity” of individuals*.

(“ambiguity” later called “diversity”)

The more **accurate** and **diverse** the individual learners,
the better the ensemble

However,

- the “ambiguity” does not have an operable definition
- The error-ambiguity decomposition is derivable only for regression setting with squared loss

很多成功的集成学习方法

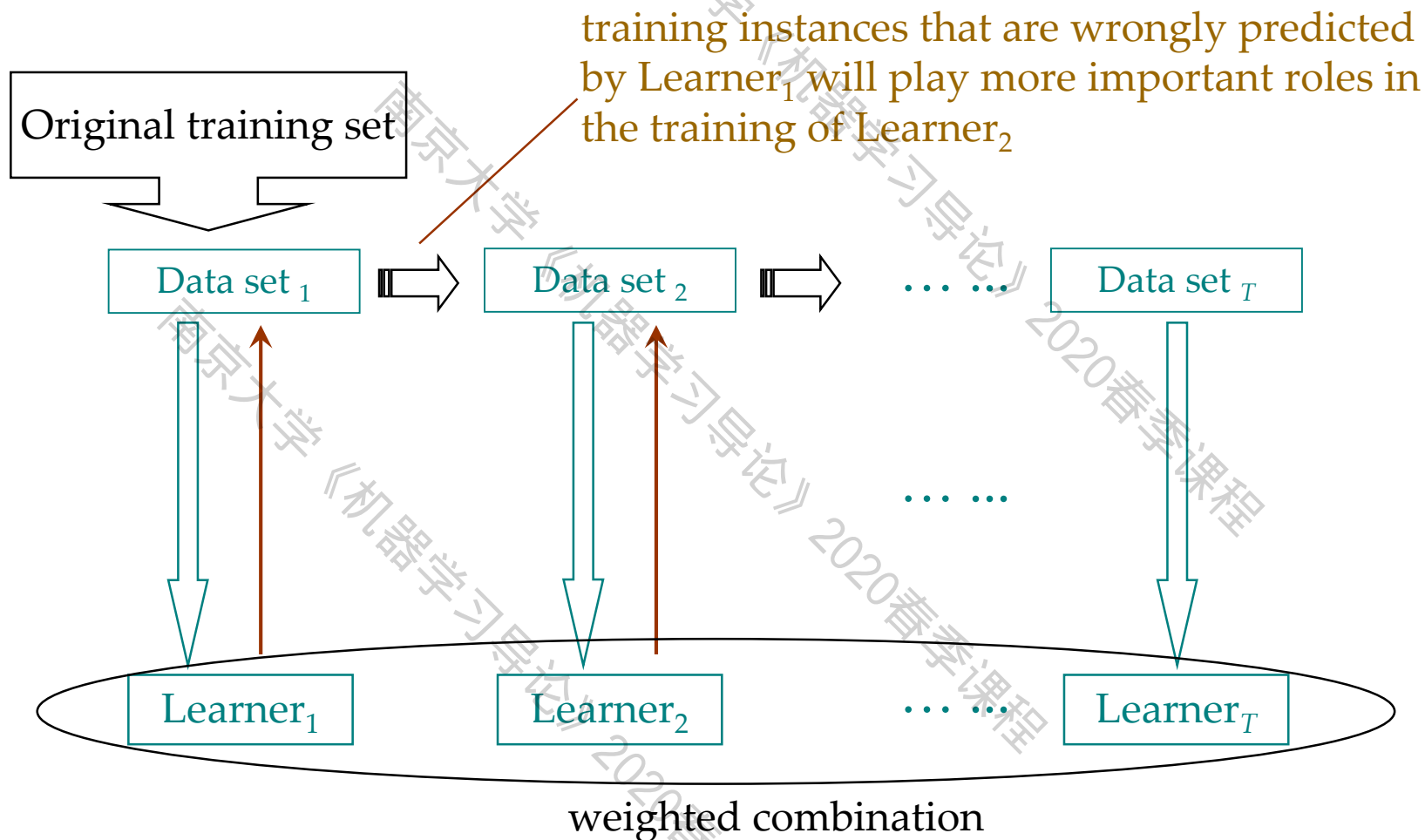
■ 序列化方法

- **AdaBoost** [Freund & Schapire, JCSS97]
- GradientBoost [Friedman, AnnStat01]
- LPBoost [Demiriz, Bennett, Shawe-Taylor, MLJ06]
-

■ 并行化方法

- **Bagging** [Breiman, MLJ96]
- Random Forest [Breiman, MLJ01]
- Random Subspace [Ho, TPAMI98]
-

Boosting: A flowchart illustration



AdaBoost

输入: 训练集 $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$;
基学习算法 \mathcal{L} ;
训练轮数 T .

过程:

1: $\mathcal{D}_1(\mathbf{x}) = 1/m$.

2: **for** $t = 1, 2, \dots, T$ **do**

3: $h_t = \mathcal{L}(D, \mathcal{D}_t)$;

4: $\epsilon_t = P_{\mathbf{x} \sim \mathcal{D}_t}(h_t(\mathbf{x}) \neq f(\mathbf{x}))$;

5: **if** $\epsilon_t > 0.5$ **then break**

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

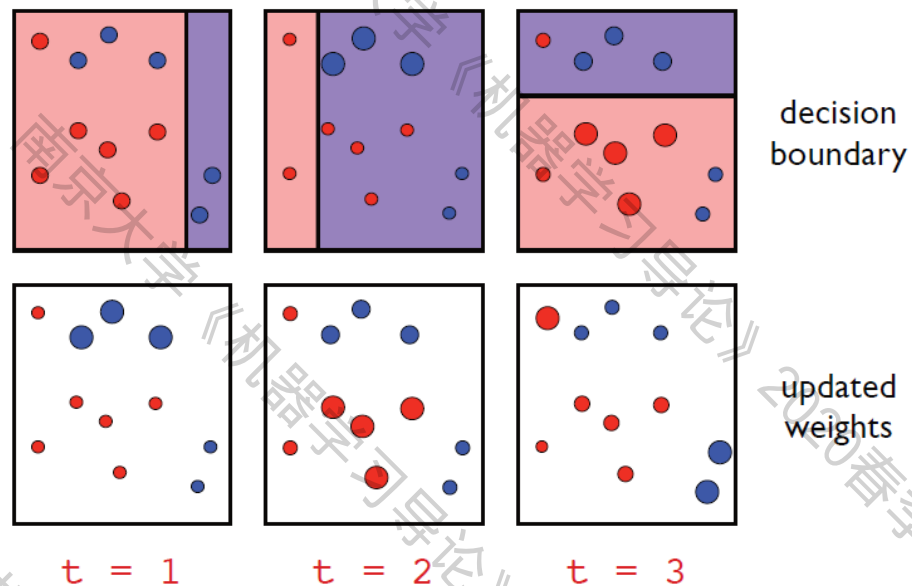
7:
$$\begin{aligned} \mathcal{D}_{t+1}(\mathbf{x}) &= \frac{\mathcal{D}_t(\mathbf{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\mathbf{x}) = f(\mathbf{x}) \\ \exp(\alpha_t), & \text{if } h_t(\mathbf{x}) \neq f(\mathbf{x}) \end{cases} \\ &= \frac{\mathcal{D}_t(\mathbf{x}) \exp(-\alpha_t f(\mathbf{x}) h_t(\mathbf{x}))}{Z_t} \end{aligned}$$

8: **end for**

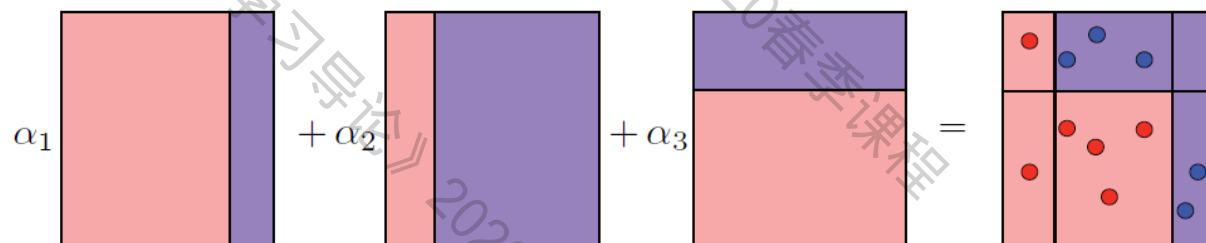
输出: $H(\mathbf{x}) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(\mathbf{x}) \right)$



AdaBoost实验

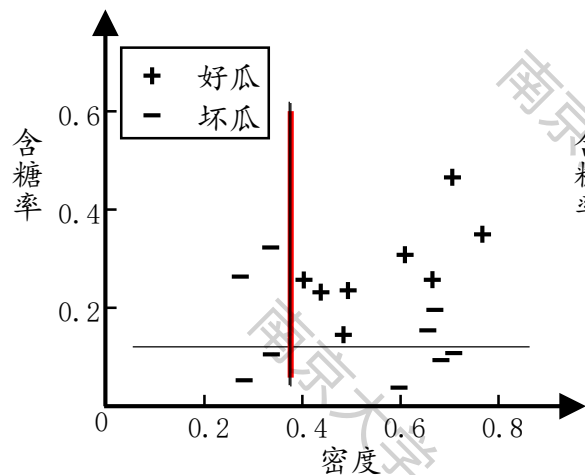


(a)

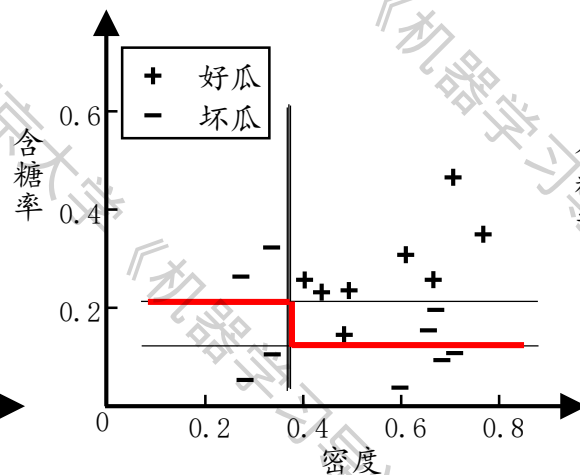


(b)

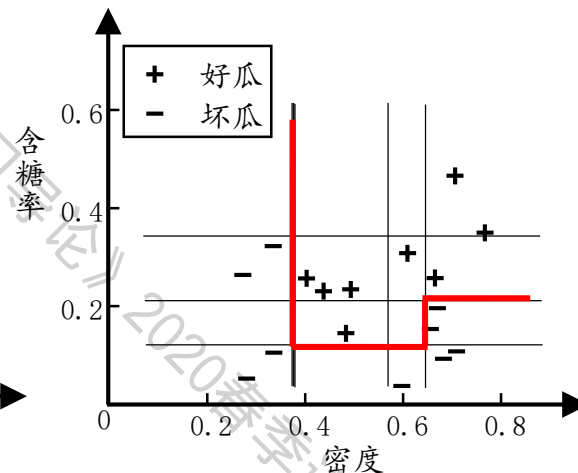
AdaBoost实验



(a) 3个基学习器



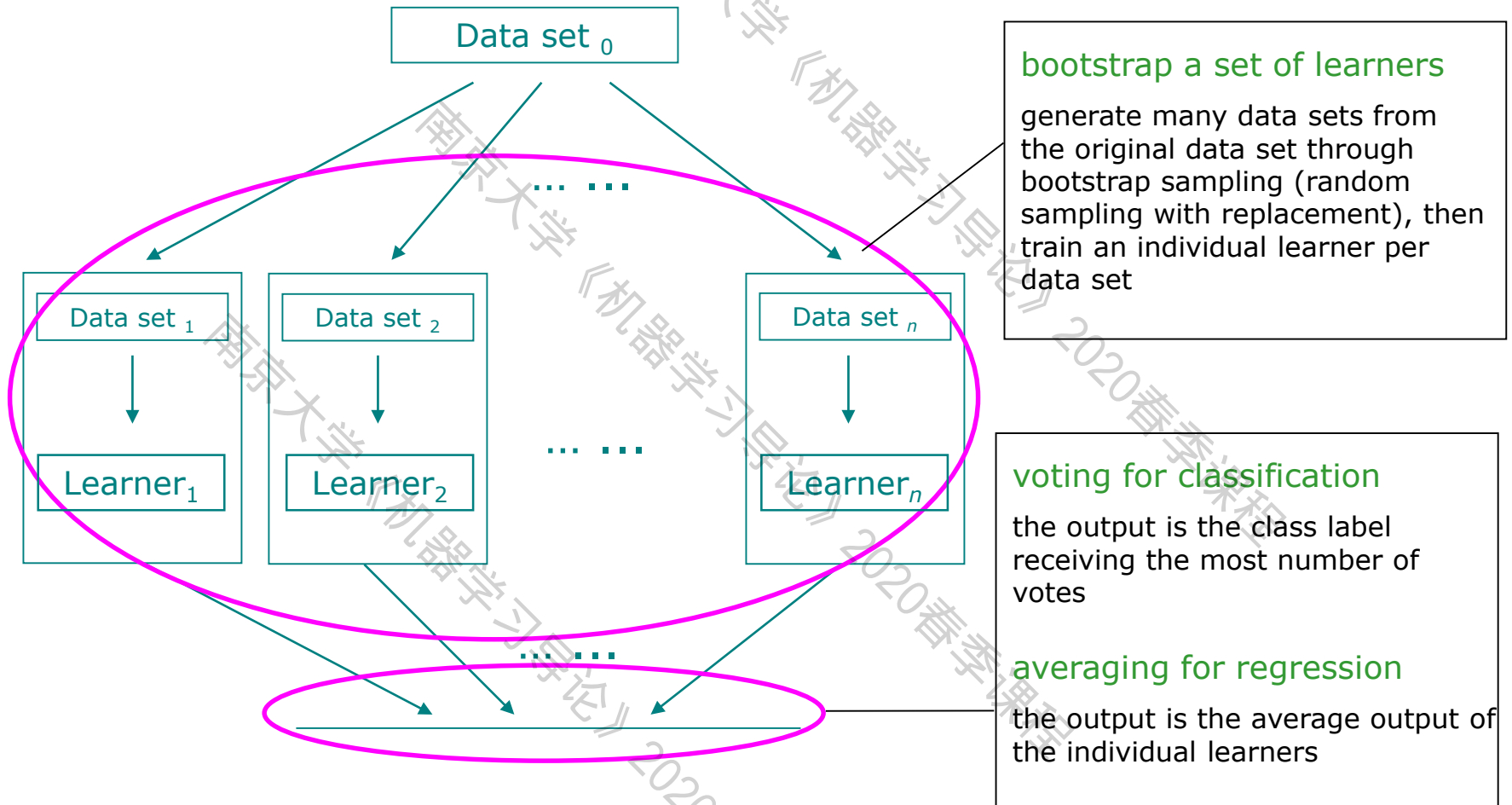
(b) 5个基学习器



(c) 11个基学习器

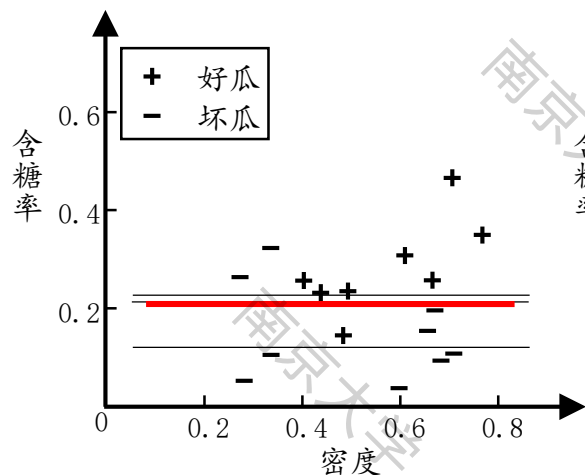
从偏差-方差的角度：降低偏差，可对泛化性能相当弱的学习器构造出很强的集成

Bagging

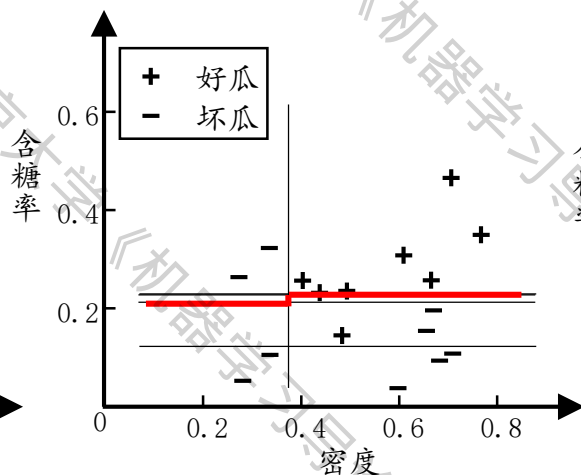




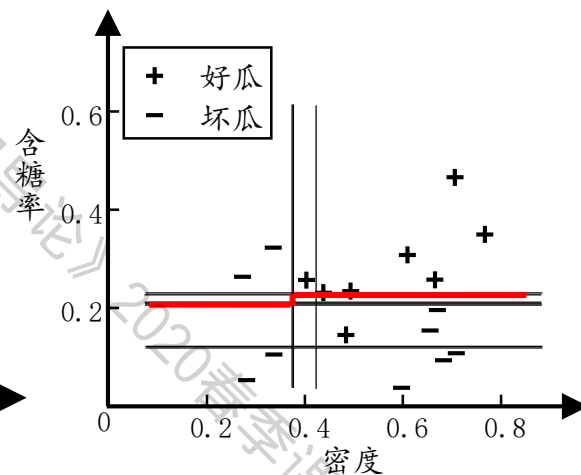
Bagging实验



(a) 3个基学习器



(b) 5个基学习器



(c) 11个基学习器

□ 从偏差-方差的角度：降低方差，在不剪枝的决策树、神经网络等易受样本影响的学习器上效果更好

学习器结合

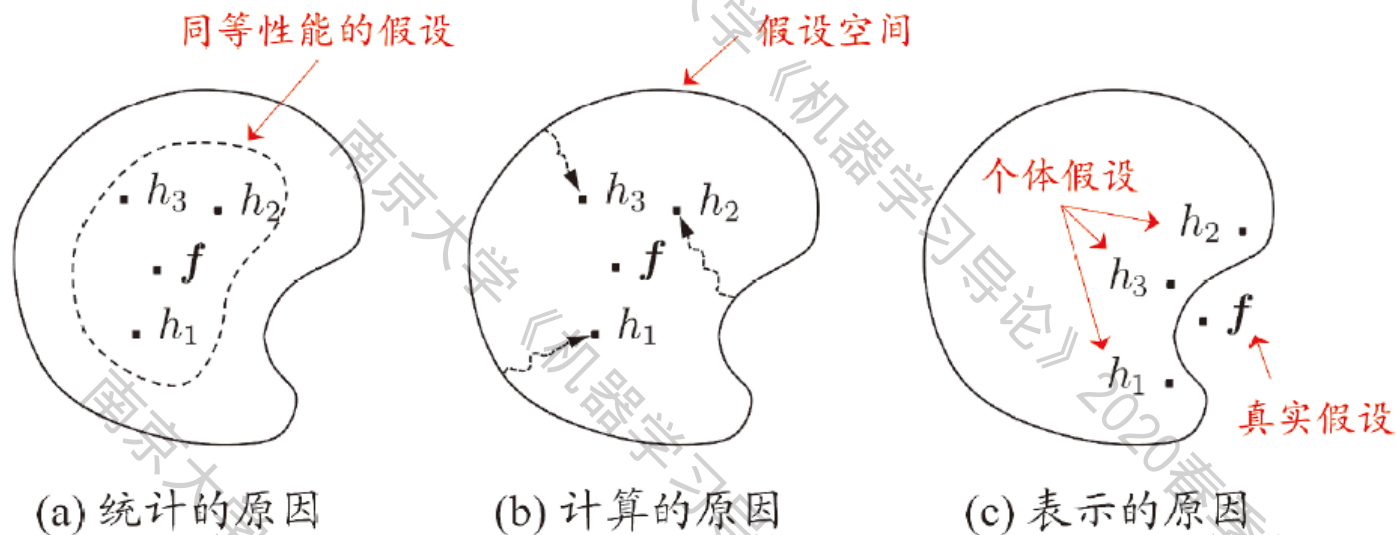


图 8.8 学习器结合可能从三个方面带来好处 [Dietterich, 2000]

常用结合方法：

▣ 投票法

- 绝对多数投票法
- 相对多数投票法
- 加权投票法

▣ 平均法

- 简单平均法
- 加权平均法

▣ 学习法

Stacking

输入: 训练集 $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$;
初级学习算法 $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_T$;
次级学习算法 \mathcal{L} .

过程:

```
1: for  $t = 1, 2, \dots, T$  do  
2:    $h_t = \mathcal{L}_t(D)$ ;  
3: end for
```

使用初级学习算法 \mathcal{L}_t
产生初级学习器 h_t .

```
4:  $D' = \emptyset$ ;
```

```
5: for  $i = 1, 2, \dots, m$  do  
6:   for  $t = 1, 2, \dots, T$  do  
7:      $z_{it} = h_t(\mathbf{x}_i)$ ;  
8:   end for
```

生成次级训练集.

```
9:    $D' = D' \cup ((z_{i1}, z_{i2}, \dots, z_{iT}), y_i)$ ;
```

```
10: end for
```

```
11:  $h' = \mathcal{L}(D')$ ;
```

输出: $H(\mathbf{x}) = h'(h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_T(\mathbf{x}))$

图 8.9 Stacking 算法

多样性

“多样性” (diversity) 是集成学习的关键

误差-分歧分解 (error-ambiguity decomposition):

$$E = \overline{E} - \overline{A}$$

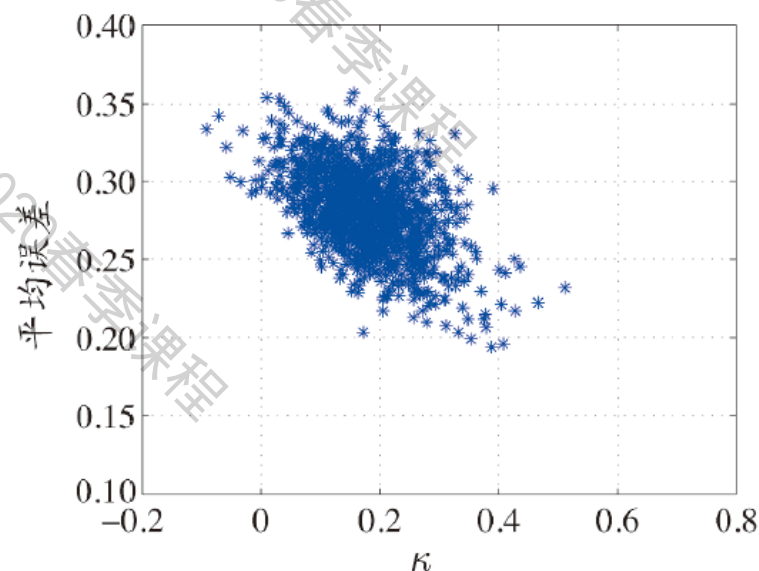
多样性度量

一般通过两分类器的预测结果列联表定义

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

- 不合度量 (disagreement measure)
- 相关系数 (correlation coefficient)
- Q-统计量 (Q-statistic)
- κ -统计量 (κ -statistic)
-

κ -误差图



每一对分类器作为图中的一个点

多样性增强常用策略

□ 数据样本扰动

- 例如 **Adaboost** 使用 重要性采样、**Bagging** 使用自助采样
- 注意：对“不稳定基学习器”（如决策树、神经网络等）很有效
不适用于“稳定基学习器”（如线性分类器、**SVM**、朴素贝叶斯等）

□ 输入属性扰动

- 例如 **随机子空间** (Random Subspace)

□ 输出表示扰动

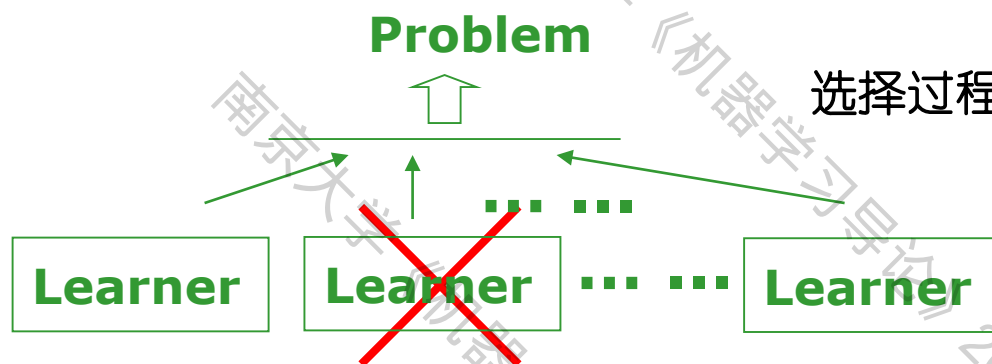
- 例如 输出标记随机翻转、分类转回归、**ECOC**

□ 算法参数扰动

“越多越好”？

选择性集成 (selective ensemble):

给定一组个体学习器，从中选择一部分来构建集成，经常会比使用所有个体学习器更好（更小的存储/时间开销，更强的泛化性能）



选择过程需考虑个体 **性能** 与 **多样性/互补性**

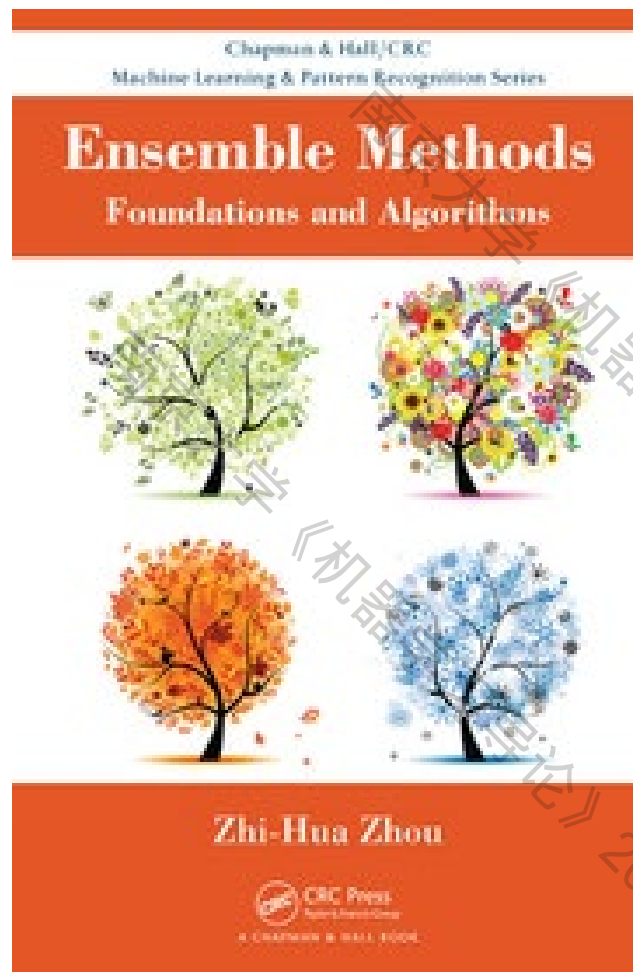
仅选出“精度最高的”通常不好！

集成修剪 (ensemble pruning)
[Margineantu & Dietterich, ICML'97]
较早出现，针对序列型集成
减小集成规模、降低泛化性能

选择性集成 [Zhou, et al, AIJ 02] 稍晚，
针对并行型集成，MCBTA (Many could
be better than all)定理
减小集成规模、增强泛化性能

目前“集成修剪”与“选择性集成”基本被视为同义词

更多关于集成学习的内容，可参考：



Z.-H. Zhou.
Ensemble Methods:
Foundations and Algorithms,
Boca Raton, FL: Chapman &
Hall/CRC, Jun. 2012.
(ISBN 978-1-439-830031)

集成学习常用软件/工具包

❑ Random Forest

<https://cran.r-project.org/web/packages/randomForest/index.html>

❑ XGBoost

<https://github.com/dmlc/xgboost>

❑ LightGBM

<https://github.com/Microsoft/LightGBM>

❑ MultiBoost (multi-class / multi-label / multi-task)

<http://www.multiboost.org/>

❑

前往.....

