八、集成学习

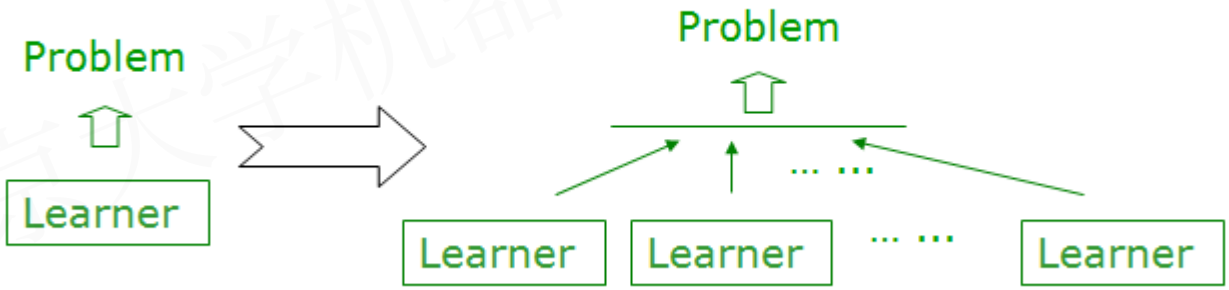
南京大学机器学习导论2023春季

主讲教师：赵鹏

集成学习

Ensemble Learning (集成学习)

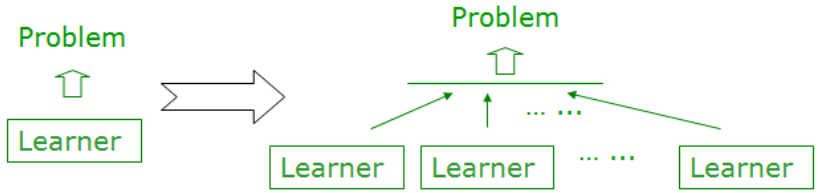
“Ensemble methods” is a machine learning paradigm where multiple (homogenous/heterogeneous) individual learners are trained for the same problem and further combined.



集成学习

Ensemble Learning (集成学习):

Using multiple learners to solve the problem



Demonstrated great performance in real practice

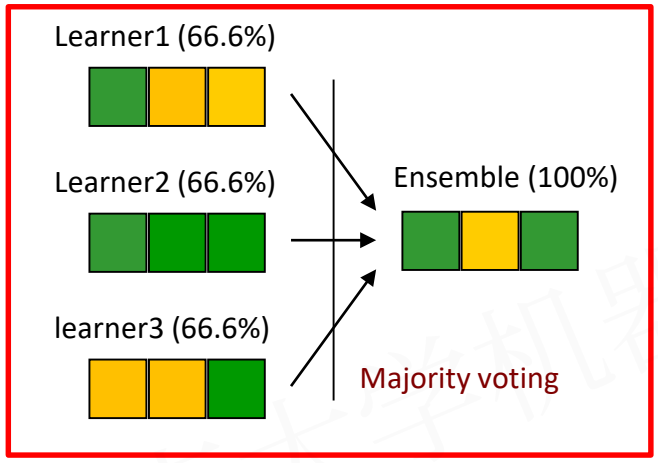
- KDDCup'07: 1st place for "... Decision Forests and ..."
- KDDCup'08: 1st place of Challenge1 for a method using Bagging; 1st place of Challenge2 for "... Using an Ensemble Method "
- KDDCup'09: 1st place of Fast Track for "Ensemble ... "; 2nd place of Fast Track for "... bagging ... boosting tree models ...", 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

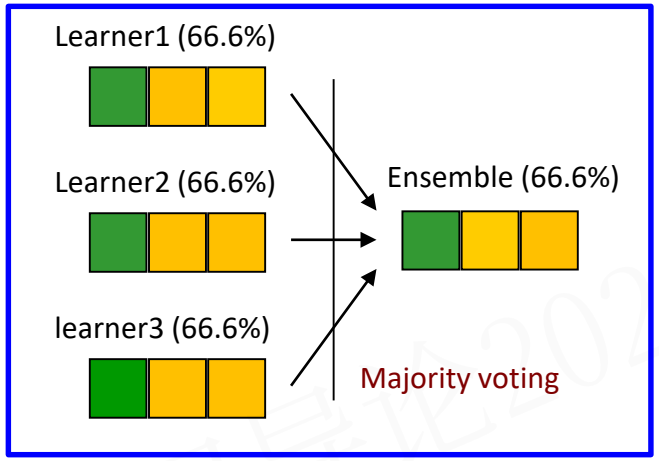
To win? Ensemble !

如何得到好的集成？

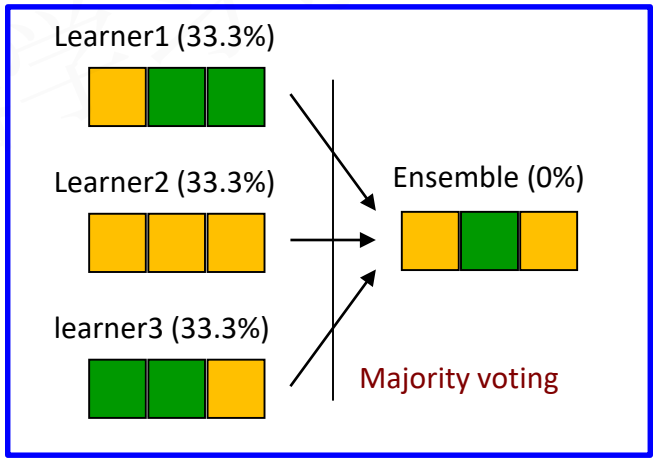
Some intuitions:



Ensemble really helps



Individuals must be different



Individuals must be not-bad

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):

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

Ensemble error — *Ave. error of individuals* — *Ave. “ambiguity” of individuals* (*“ambiguity” later called “diversity”*)

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

However,

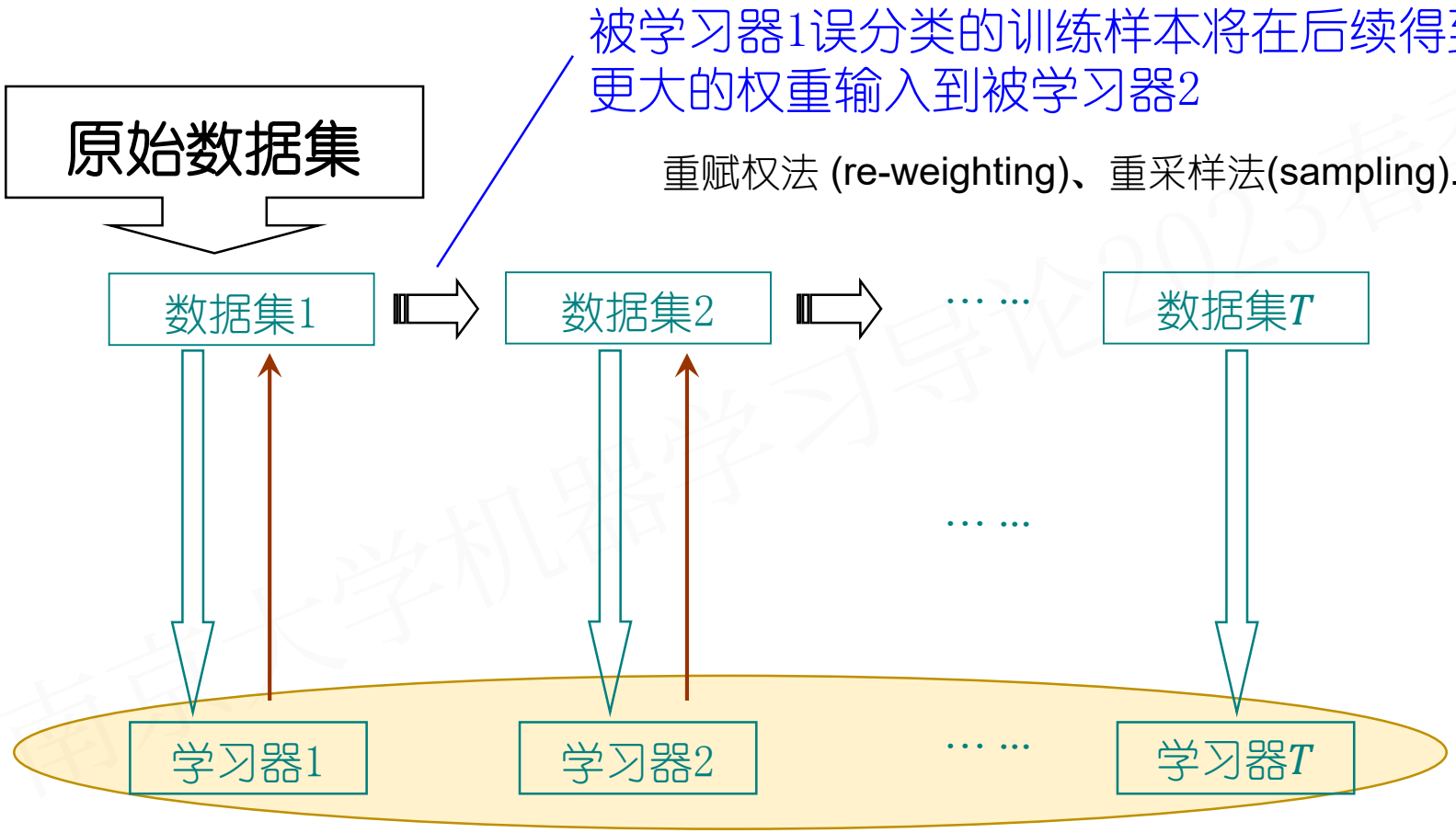
- the “ambiguity” does not have an operable definition
- The error-ambiguity decomposition is derivable only for regression setting with squared loss

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]
  - … …

南京大学机器学习导论2023春季

Boosting：流程示意图



AdaBoost：算法流程

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： $F(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \right)$

被误分类的样本权重增加，使基学习器进一步关注到这些“困难”训练样本

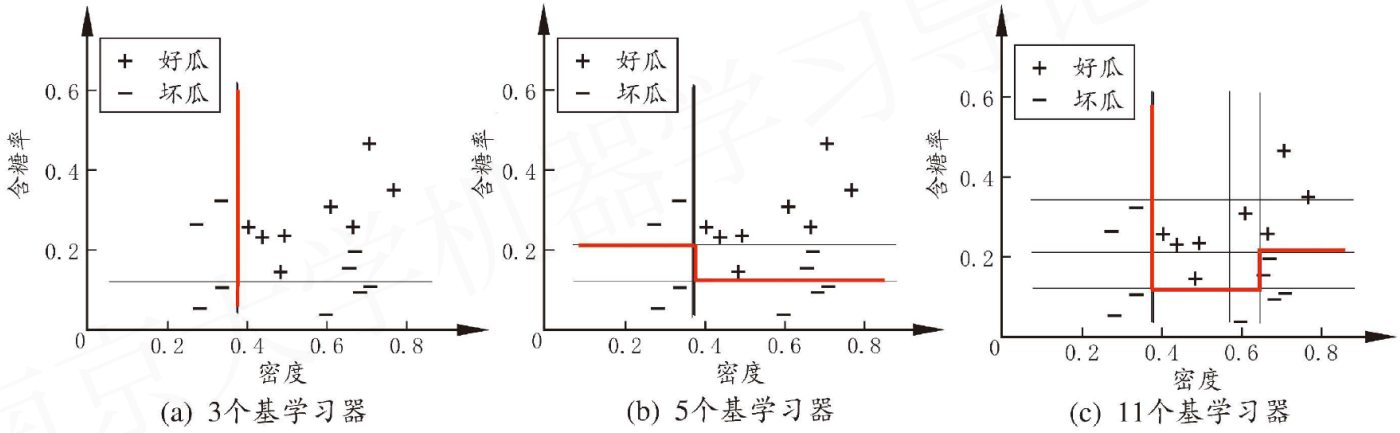
最终分类器加权结合，误差越低，权重越大

南京大学机器学习课程2023春季

AdaBoost

AdaBoost推导：加性模型、在线学习、贝叶斯模型平均等

Boosting方法：有效降低偏差，将泛化性能弱的分类器提升到很强的集成。



(a) 3个基学习器　(b) 5个基学习器　(c) 11个基学习器

图 8.4 西瓜数据集 3.0α 上 AdaBoost 集成规模为 3、5、11 时，集成(红色)与基学习器(黑色)的分类边界.

AdaBoost

Yoav Freund

Robert Schapire

Goldel Prize 2003

This paper introduced AdaBoost, an adaptive algorithm to improve the accuracy of hypotheses in machine learning. The algorithm demonstrated novel possibilities in analysing data and is a permanent contribution to science even beyond computer science.

JOURNAL OF COMPUTER AND SYSTEM SCIENCES 55, 119–139 (1997)
ARTICLE NO. SS971504

A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*

Yoav Freund and Robert E. Schapire†

AT&T Labs, 180 Park Avenue, Florham Park, New Jersey 07932

Received December 19, 1996

In the first part of the paper we consider the problem of dynamically apportioning resources among a set of options in a worst-case on-line framework. The model we study can be interpreted as a broad, abstract extension of the well-studied on-line prediction model to a general decision-theoretic setting. We show that the multiplicative weight-update Littlestone–Warmuth rule can be adapted to this model, yielding bounds that are slightly weaker in some cases, but applicable to a considerably more general class of learning problems. We show how the resulting learning algorithm can be applied to a variety of problems, including gambling, multiple-outcome prediction, repeated games, and prediction of points in $\mathbb{R}^n$. In the second part of the paper we apply the multiplicative weight-update technique to derive a new boosting algorithm. This boosting algorithm does not require any prior knowledge about the performance of the weak learning algorithm. We also study generalizations of the new boosting algorithm to the problem of learning functions whose range, rather than being binary, is an arbitrary finite set or a bounded segment of the real line. © 1997 Academic Press

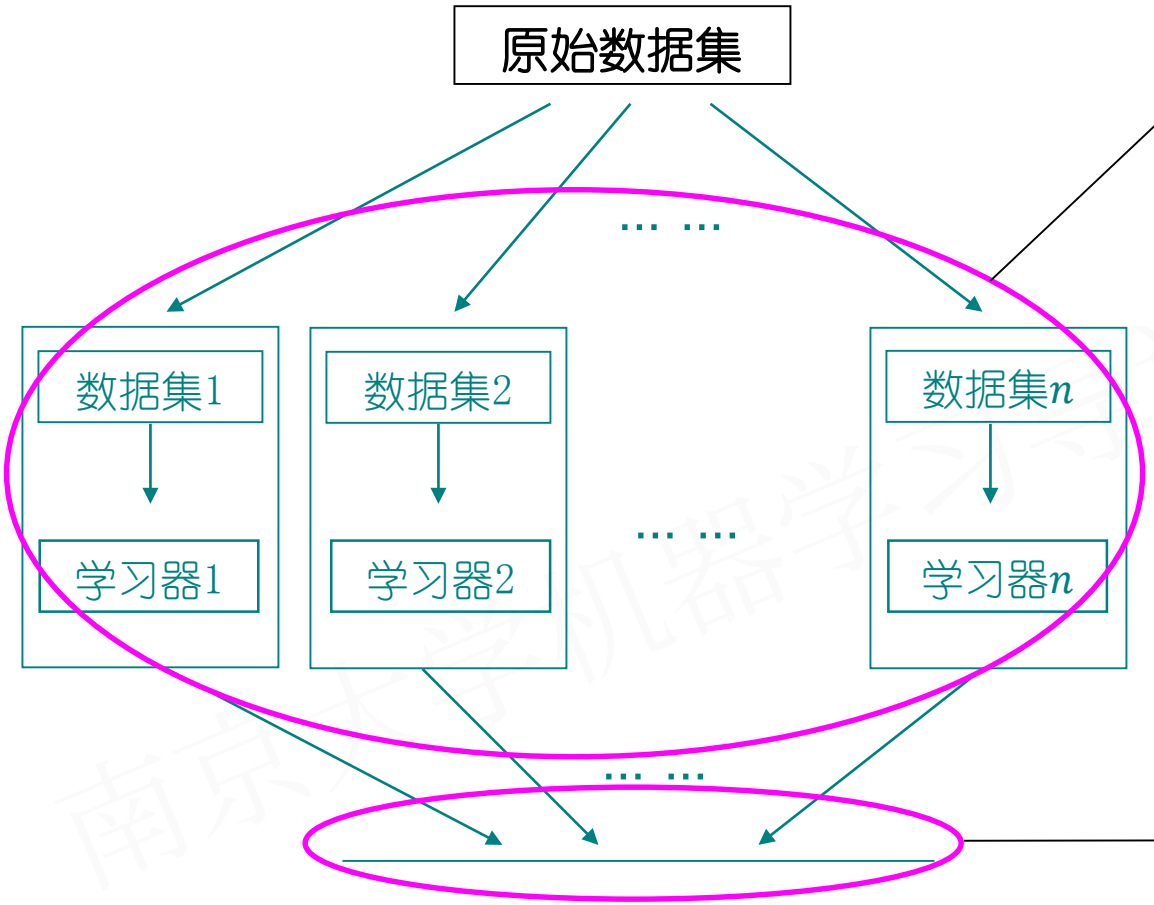
converting a "weak" PAC learning algorithm that performs just slightly better than random guessing into one with arbitrarily high accuracy.

We formalize our *on-line allocation model* as follows. The allocation agent A has N options or *strategies* to choose from; we number these using the integers $1, \ldots, N$. At each time step $t = 1, 2, \ldots, T$, the allocator A decides on a distribution $\mathbf{p}^t$ over the strategies; that is $p_i^t \geqslant 0$ is the amount allocated to strategy i, and $\sum_{i=1}^{N} p_i^t = 1$. Each strategy i then suffers some *loss* ℓ_i^t which is determined by the (possibly adversarial) "environment." The loss suffered by A is then $\sum_{i=1}^{n} p_i^t \ell_i^t = \mathbf{p}^t \cdot \boldsymbol{\ell}^t$, i.e., the average loss of the strategies with respect to A's chosen allocation rule. We call this loss function the *mixture loss*.

In this paper, we always assume that the loss suffered by any strategy is bounded so that, without loss of generality, $\ell_i^t \in [0, 1]$. Besides this condition, we make no assumptions

Reference: Y. Freund and R. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. JCSS 1997.

Bagging (Booststrap AGGregatING)



bootstrap a set of learners

基于原始数据集，通过bootstrap（有放回随机采样）产生多个子数据集；然后基于每个数据集训练得到一个学习器

voting for classification

Bagging最终输出标签为得到最多学习器投票的标签

averaging for regression

Bagging最终输出预测为多个学习器输出的平均

Bagging

- 较为一般化，不经修改用于多分类、回归等任务
- 每个基学习器只用了初始训练集中约63.2%的样本，剩下约36.8%的样本可以做验证集进行“包外估计”

Bagging方法：有效降低方差，对易受到样本干扰的学习上效用更为明显。

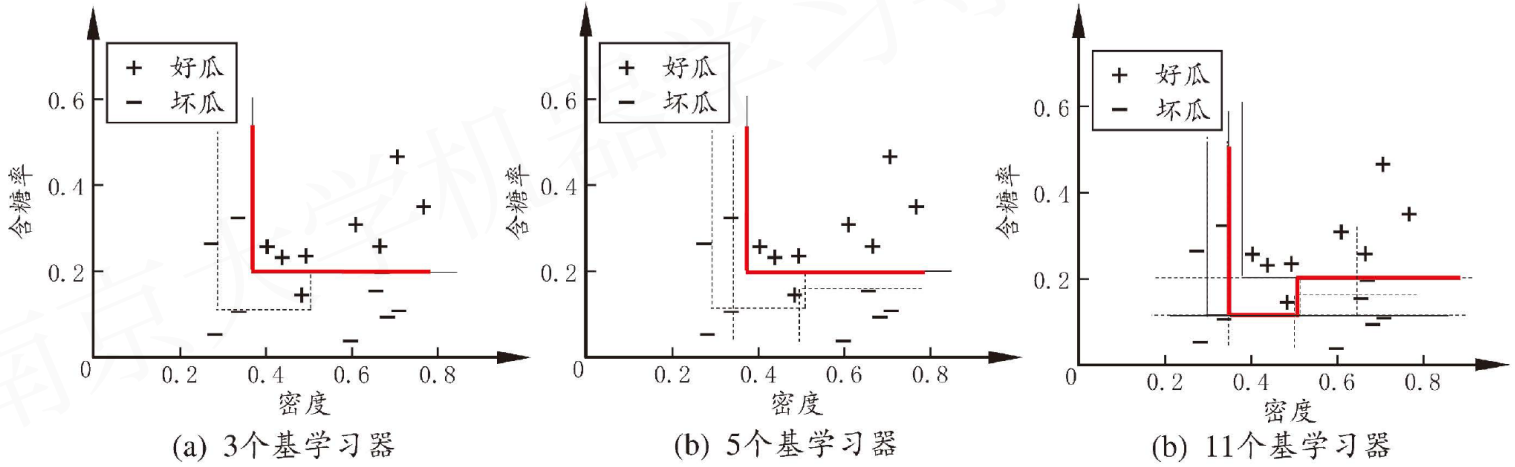


图 8.6 西瓜数据集 3.0α 上 Bagging 集成规模为 3、5、11 时，集成(红色)与基学习器(黑色)的分类边界.

随机森林 (RF, random forest)

以**决策树**为基学习器，一种**Bagging**的变种

核心：**随机属性选择**

1）基决策树的每个节点，先从节点属性集合中随机选择一个包含k个属性的子集

2）再从这个子集中选择一个最优属性进行划分

$k = 1$：随机选择一个属性用于划分

$k = d$：与传统决策树的构建相同

一般情况下，推荐值为：$k = \log_2 d$

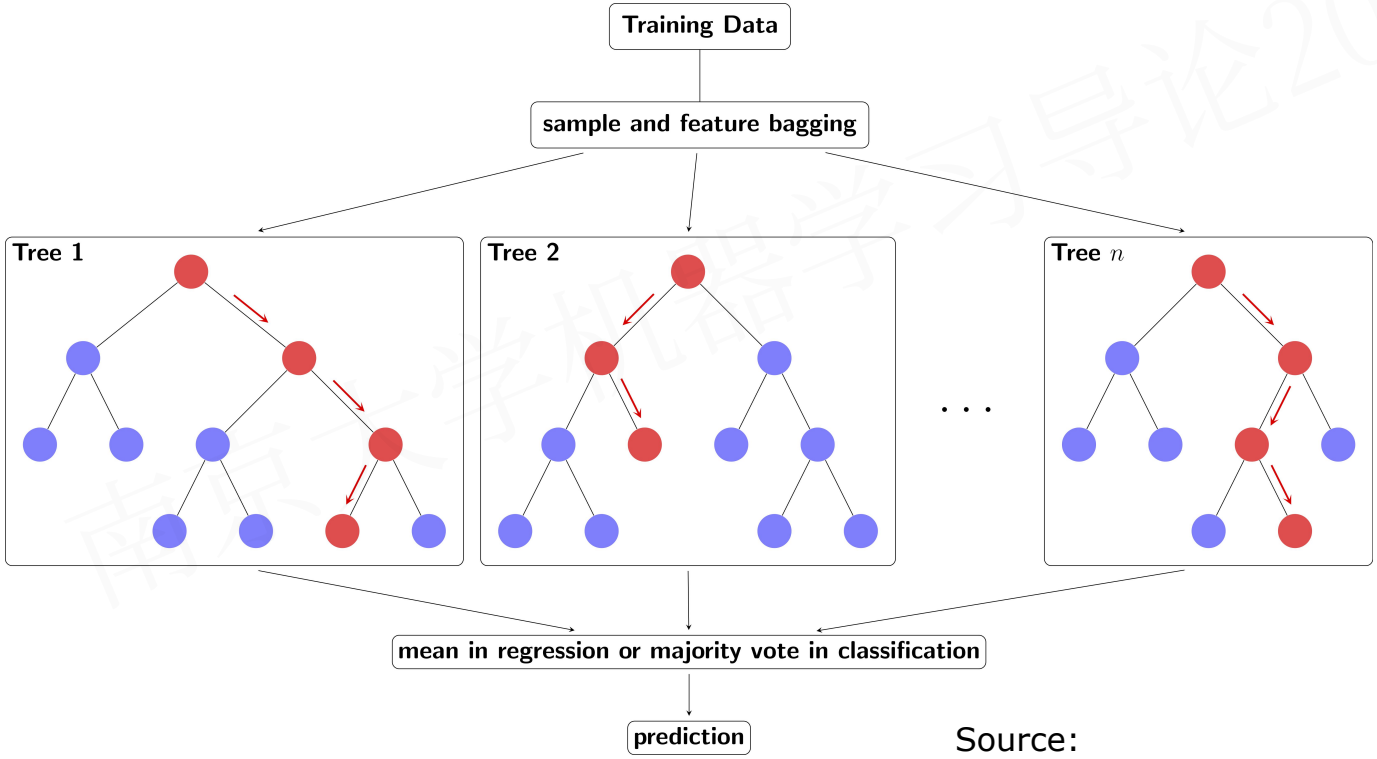
南京大学机器学习导论2023春季

随机森林 (RF, random forest)

核心：**随机属性选择**

1) 基决策树的每个节点，先从节点属性集合中随机选择一个包含k个属性的子集

2) 再从这个子集中选择一个最优属性进行划分



Source:
https://tex.stackexchange.com/questions/503883

南京大学机器学习导论2023春季

Bagging & RF

Leo Breiman

(1928 – 2005)

1996 Machine Learning

Bagging Predictors

LEO BREIMAN leo@stat.berkeley.edu

Statistics Department, University of California, Berkeley, CA 94720

Editor: Ross Quinlan

Abstract. Bagging predictors is a method for generating multiple versions of a predictor and using these to get an aggregated predictor. The aggregation averages over the versions when predicting a numerical outcome and does a plurality vote when predicting a class. The multiple versions are formed by making bootstrap replicates of the learning set and using these as new learning sets. Tests on real and simulated data sets using classification and regression trees and subset selection in linear regression show that bagging can give substantial gains in accuracy. The vital element is the instability of the prediction method. If perturbing the learning set can cause significant changes in the predictor constructed, then bagging can improve accuracy.

Keywords: Aggregation, Bootstrap, Averaging, Combining

2001 Machine Learning

Random Forests

LEO BREIMAN

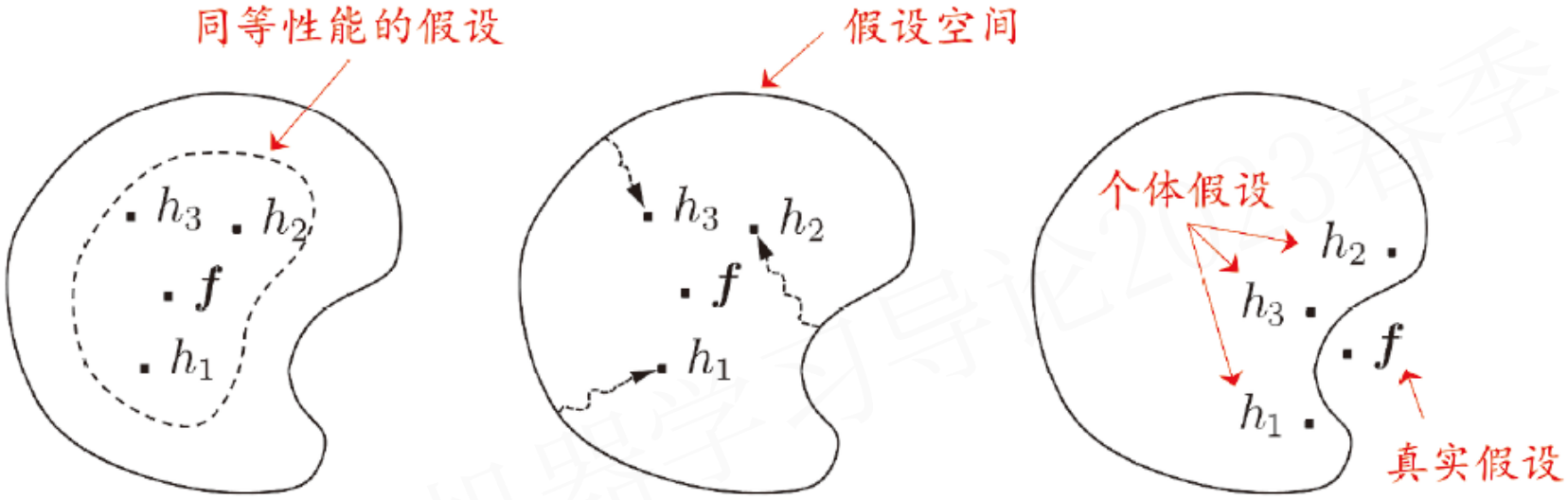
Statistics Department, University of California, Berkeley, CA 94720

Editor: Robert E. Schapire

Abstract. Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges a.s. to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. Using a random selection of features to split each node yields error rates that compare favorably to Adaboost (Y. Freund & R. Schapire, *Machine Learning*: *Proceedings of the Thirteenth International conference*, ∗ ∗ ∗, 148–156), but are more robust with respect to noise. Internal estimates monitor error, strength, and correlation and these are used to show the response to increasing the number of features used in the splitting. Internal estimates are also used to measure variable importance. These ideas are also applicable to regression.

Keywords: classification, regression, ensemble

学习器结合



(a) 统计的原因　(b) 计算的原因　(c) 表示的原因

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

学习器结合

常用结合方法：

- 平均法 (averaging)
  - 简单平均法

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

  - 加权平均法

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

其中 w_i 是个体学习器 h_i 的权重, 通常要求 $w_i \geqslant 0, \sum_{i=1}^{T} w_i = 1.$

南京大学机器学习导论2023春季

学习器结合

常用结合方法：

- 投票法 (voting)
  - 绝对多数投票法 (majority voting)

$$H(\boldsymbol{x}) = \begin{cases} c_j, & \text{if } \sum_{i=1}^{T} h_i^j(\boldsymbol{x}) > 0.5 \sum_{k=1}^{N} \sum_{i=1}^{T} h_i^k(\boldsymbol{x}) \\ \text{reject}, & \text{otherwise} \end{cases}$$

即若某标记得票过半数，则预测为该标记；否则拒绝预测.

南京大学机器学习导论2023春季

学习器结合

常用结合方法：

- 投票法 (voting)
  - 相对多数投票法 (plurality voting)

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

即预测为得票最多的标记；若同时多个标记获最高票，则从中随机选一个.

  - 加权投票法 (weighted voting)

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

其中 w_i 是个体学习器 h_i 的权重, 通常要求 $w_i \geqslant 0, \sum_{i=1}^{T} w_i = 1.$

南京大学机器学习导论2023春季

投票法：注意事项

以“加权投票法”为例

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

- 类标记：$h_i^j(\boldsymbol{x}) \in \{0,1\}$，相当于使用类标记投票，又称为“硬投票”
- 类概率：$h_i^j(\boldsymbol{x}) \in [0,1]$，相当于使用类概率投票，又称为“软投票”

需注意：不同学习器的输出未必可比！

“校准” (calibration) 操作：
Platt 缩放(Platt scaling) 、等分回归(isotonic regression) 等

南京大学机器学习导论2023春季

学习器结合

常用结合方法：

- 学习法 (learning-based method)

当训练数据很多时，一种更为强大的结合策略是使用“学习法”，即通过另一个学习器来进行结合.

Stacking 算法

1）先从初始数据集训练出初级学习器
2）进而“生成”一个新数据集用于训练次级学习器

新数据集中，初级学习器的输出被当作样例输入特征，而初始样本的标记仍被当作样例标记.

南京大学机器学习导论2023春季

Stacking

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

过程：

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad h_t = \mathfrak{L}_t(D)$;
3: **end for**

使用初级学习算法 $\mathfrak{L}_t$ 产生初级学习器 h_t.

4: $D' = \varnothing$;
5: **for** $i = 1, 2, \ldots, m$ **do**
6: $\quad$ **for** $t = 1, 2, \ldots, T$ **do**
7: $\quad\quad z_{it} = h_t(\boldsymbol{x}_i)$;
8: $\quad$ **end for**
9: $\quad D' = D' \cup ((z_{i1}, z_{i2}, \ldots, z_{iT}), y_i)$;
10: **end for**

生成次级训练集.

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

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

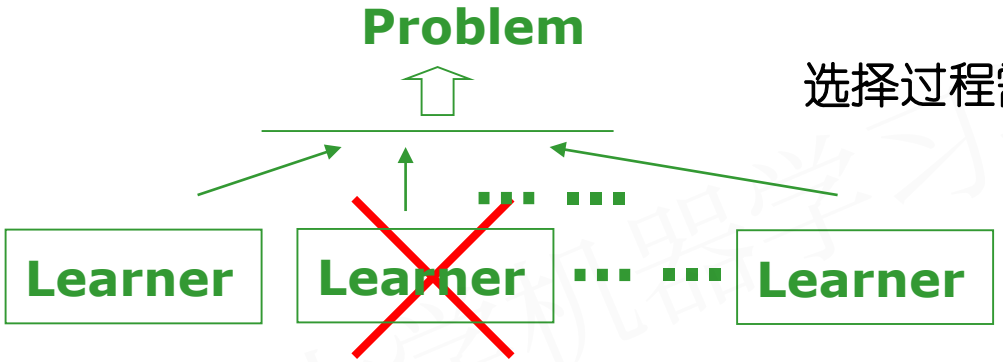
图 8.9 Stacking 算法

南京大学《机器学习导论》2023春季

“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

多样性

“多样性” (diversity) 是集成学习的关键

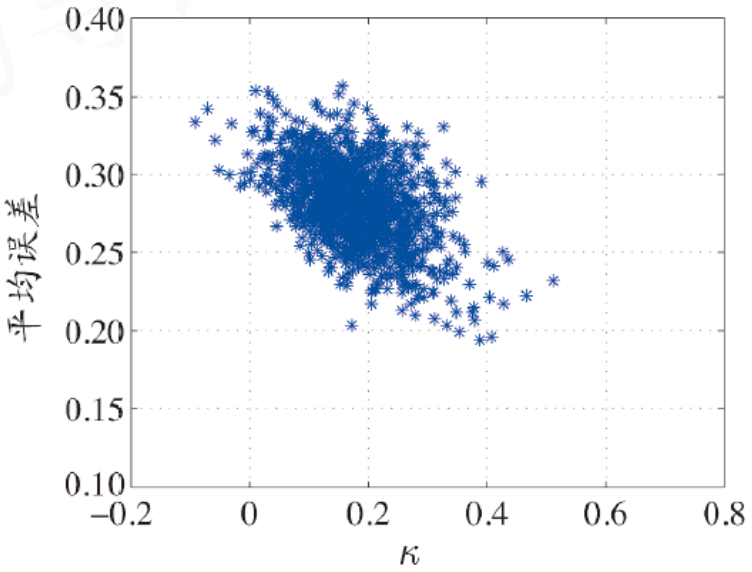
多样性度量

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)
- ……

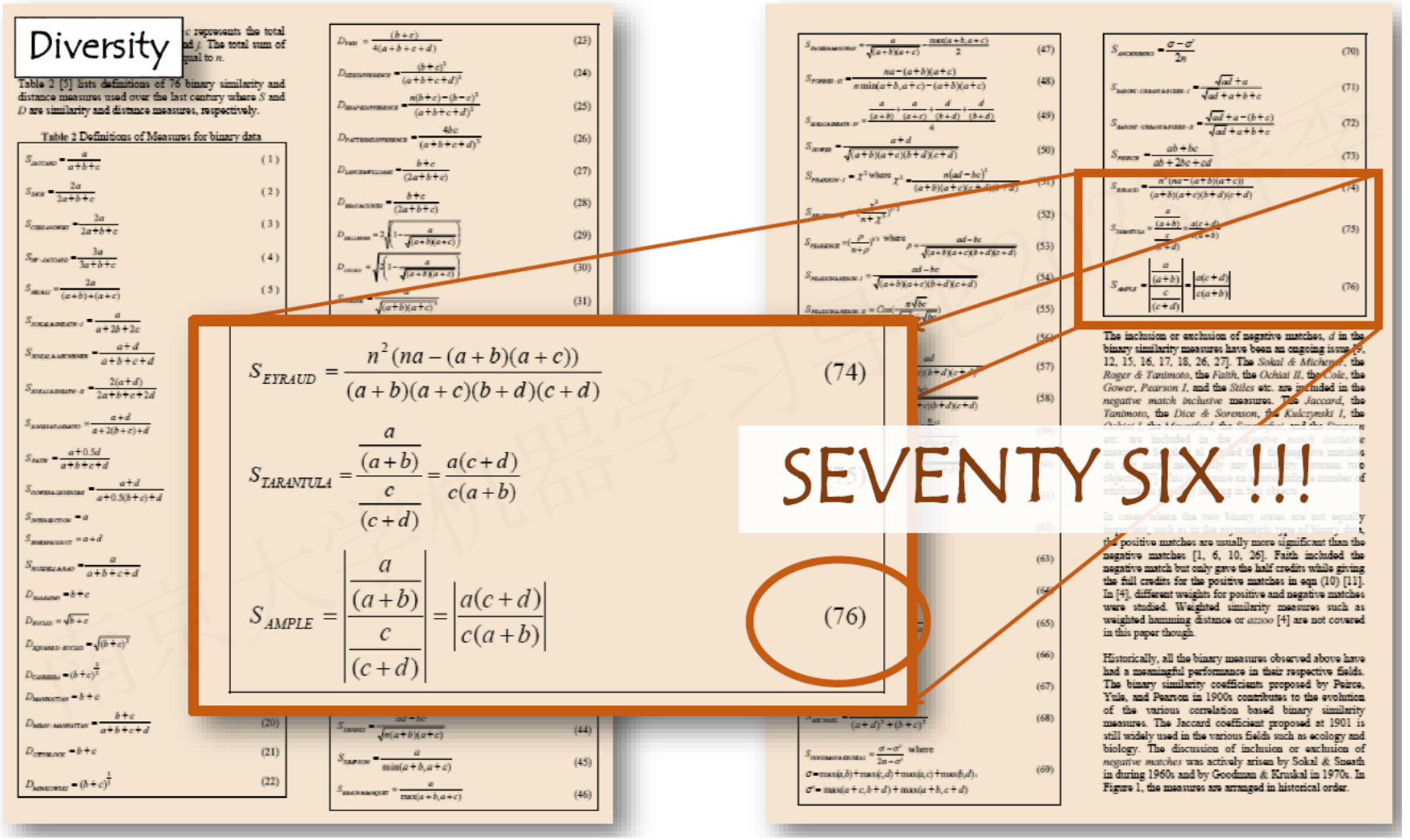
κ-误差图



每一对分类器作为图中的一个点

南京大学机器学习导论2023春季

研究者提出了很多 Diversity measure



From [L. Kuncheva, ICPRAM'16 keynote]

However, …

- [Kuncheva & Whitaker, MLJ 2003]: Empirical study shows that there seems no clear relation between many diversity measures and the ensemble performance
- [Tang, Suganthan, Yao, MLJ 2006]: Exploiting many diversity measures explicitly is ineffective in constructing consistently stronger ensembles

There is no well-accepted definition/formulation of diversity

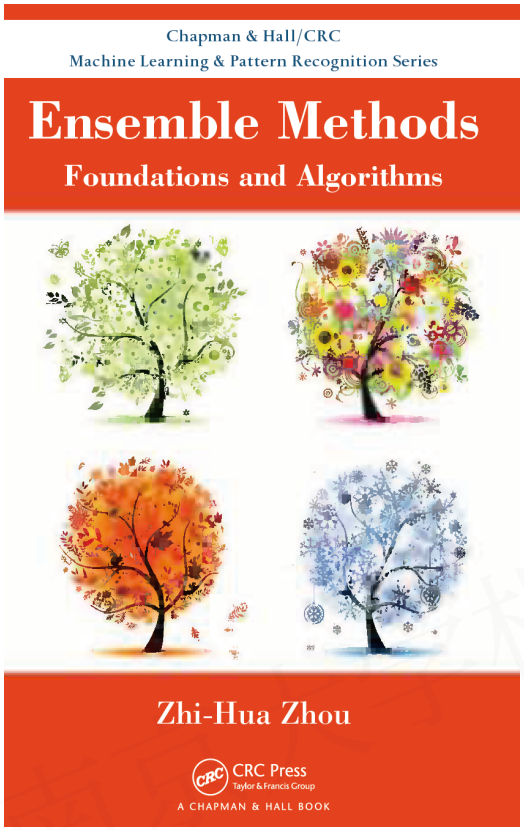
"What is diversity" remains the holy grail problem of ensemble learning

多样性增强常用策略

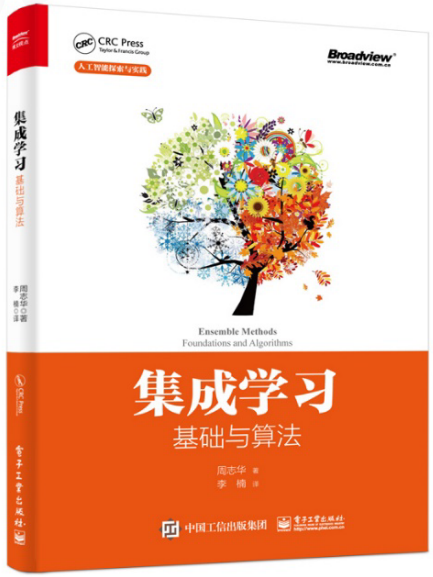
- 数据样本扰动
  - 例如 Adaboost 使用 重要性采样、Bagging 使用自助采样
  - 注意：对“不稳定基学习器”（如决策树、神经网络等）很有效
    不适用于“稳定基学习器”（如线性分类器、SVM、朴素贝叶斯等）
- 输入属性扰动
  - 例如 随机子空间 (Random Subspace)
- 输出表示扰动
  - 例如 输出标记随机翻转、分类转回归、ECOC
- 算法参数扰动

南京大学机器学习导论2023春季

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)



前往下一站...