

回顾

- 贝叶斯分类器的判定规则是什么？
- 朴素贝叶斯分类的“朴素”之处在于做了什么假设？
- 朴素贝叶斯分类中，离散属性和连续属性的 $P(x_i|c)$ 分别怎么计算？
- 为什么要进行拉普拉斯修正？如何修正？
- 如果数据集相对固定且对预测速度要求较高，应该采用哪种模式的朴素贝叶斯算法？

第四章 分类

4.1 模型评估和性能度量

4.2 k最近邻分类

4.3 决策树

4.4 贝叶斯分类

4.5 组合分类

4.6 案例：信用违约预测

组合分类

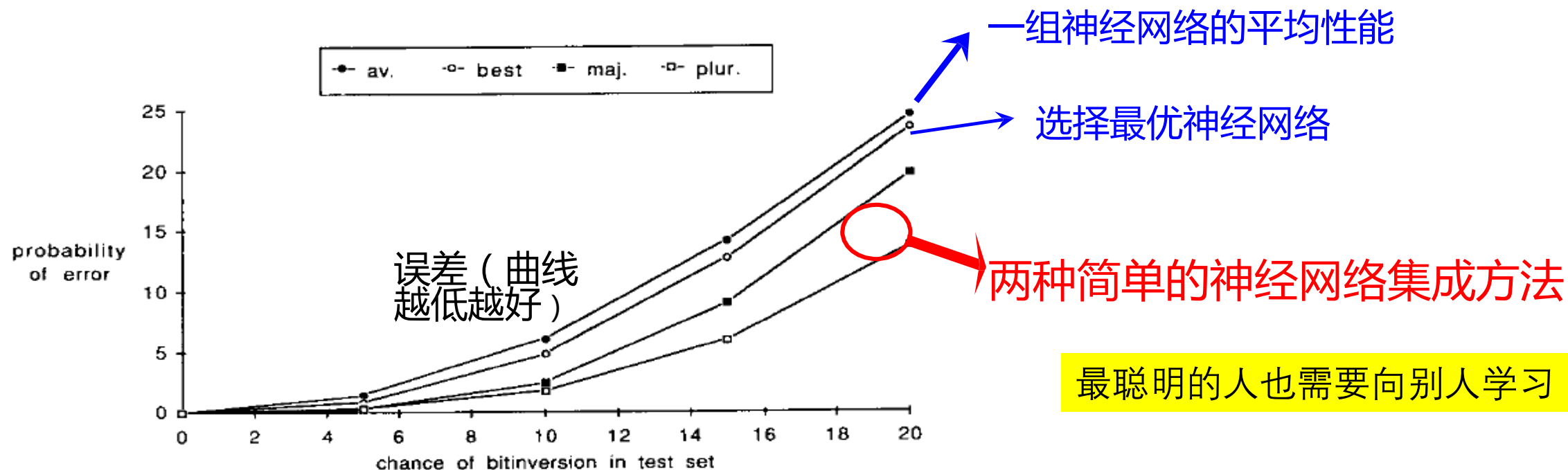
对同一个数据集，不同的分类器往往产生不同的结果。

组合分类利用多个分类器之间的差异，把多个基分类器 (base classifier) 组合得到泛化能力更高的模型。

这是一种进一步提升模型泛化能力的常用方法。在机器学习中又叫集成学习 (ensemble learning)。

为什么要集成?

组合的泛化性能通常显著优于单个学习器的泛化性能



最聪明的人也需要向别人学习

Fig. 4. Performance versus noise level in the test set is shown for individual and for consensus decisions. Data displayed shows the average and the best network, as well as collective decisions using majority and plurality for seven networks trained on individual training sets.

IEEE Transactions on **P**attern
Analysis and **M**achine Intelligence
(CCF A类期刊)

[Hansen & Salamon, TPAMI90]

浙江工业大学-计算机科学与技术学院

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”

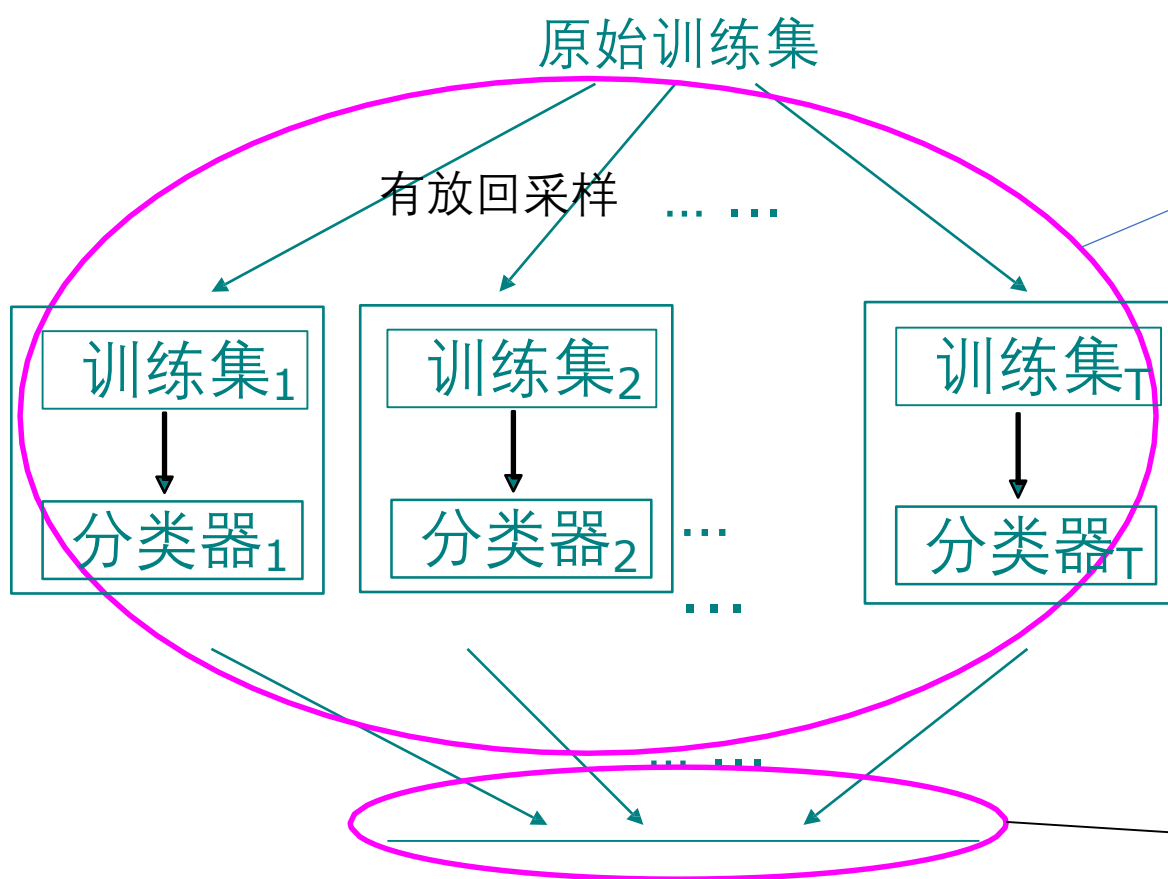
现实各类机器学习、数据挖掘应用中，广泛使用集成学习技术

想获胜，用集成

很多成功的集成学习方法

- 并行化方法：装袋 (Bagging)
 - 随机森林 (**Random Forest**)
 - 随机子空间 (Random Subspace)
 -
- 序列化方法：提升 (Boosting)
 - **AdaBoost**
 - GradientBoost
 - LPBoost
 -

Bagging框架



训练阶段:

通过有放回随机采样 (自助法 bootstrap) 得到多个数据集, 用于训练独立的基分类器。

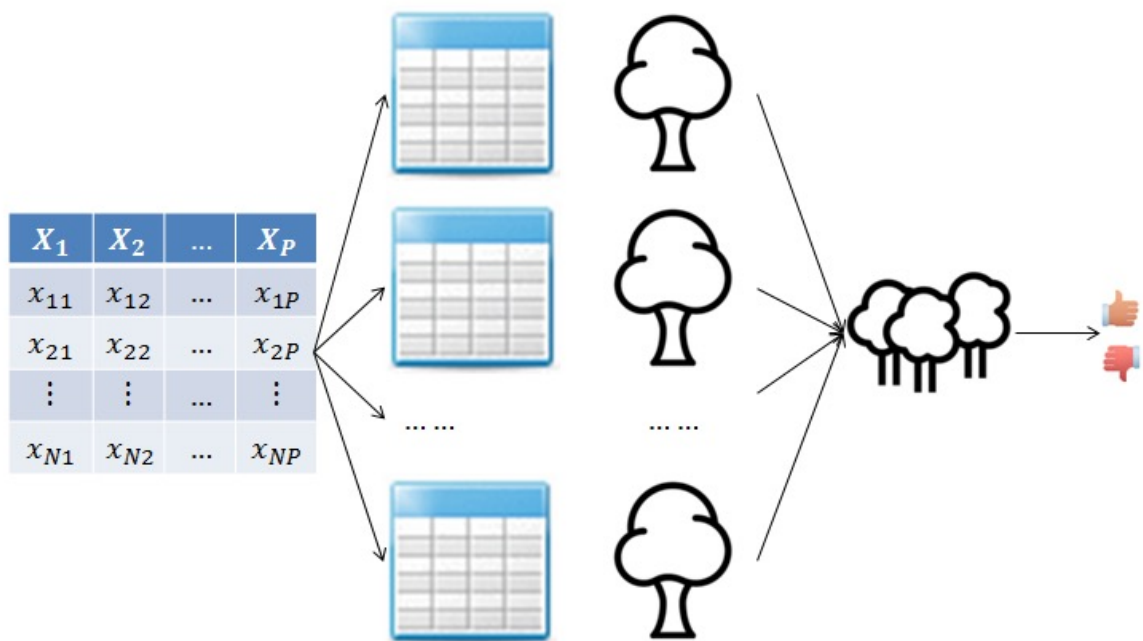
目的: 增加基分类器多样性

预测阶段:

投票决定最后的分类结果

- 预测为获得票数最多的类

Bagging实例：随机森林



随机森林：
Bagging框架下以决策树为基分类器的组合分类

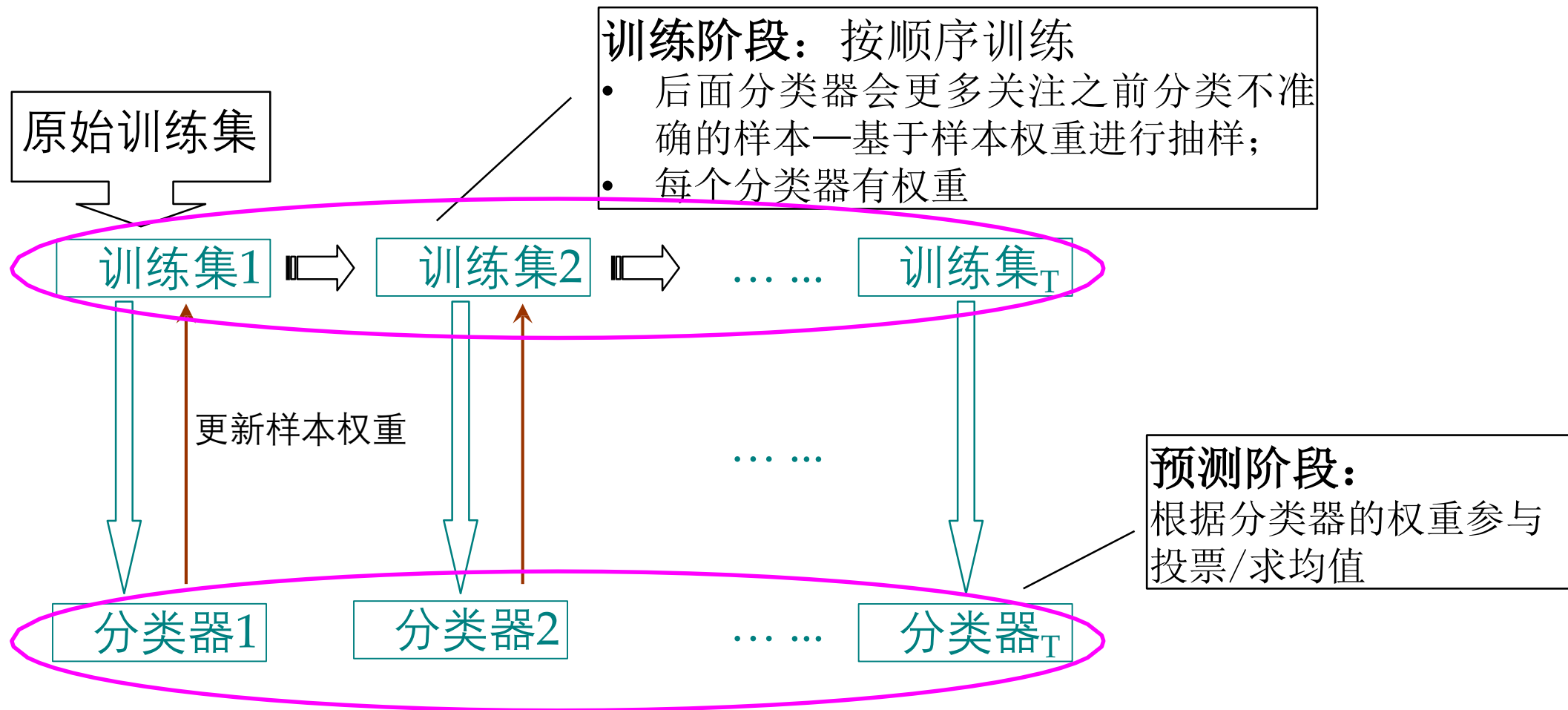
1. 利用有放回采样，从原始数据集中生成 k 个数据集，每个数据集都含有 N 个样本和 P 个特征（即与给定数据集规模一样）。
2. 让每棵决策树尽可能地充分生长，使得树中的每个节点尽可能“纯净”，即随机森林中的每棵子树都不进行剪枝。
3. 针对 k 棵CART树的随机森林，对分类问题利用投票法，将最高得票的类别用于最终的判断结果；对回归问题利用均值法，将其用作预测样本的最终结果。

随机训练集 vs 随机属性集

如何增加基分类器之间的多样性？

- 用不同**训练（样本）集**，如自助法Bootstrap，对行进行采样。
- 用不同的**属性集**，每个分类器训练时只看到部分属性-对列进行采样。
 - ✓ 对高维数据，即原始属性个数很大的情况，仅采样部分属性(子空间)可以提高每个基分类器的训练速度。
 - ✓ 对原属性数目较少时，可以采用对原属性的线性变换得到新的属性子集。比如对原属性进行多次随机线性组合产生新的属性集。
- 同时对样本和属性采样以增加分类器多样性。

Boosting: 框架



AdaBoost (Adaptive Boost)

- 样本有不同的权重 w_i , 权重越大越可能出现在下一个分类器的训练集
- 分类器错误率 $err(C_t)$ (若 $err(C_t) > 0.5$, 则丢弃)

$$err(C_t) = \sum_{y'_i \neq y_i} w_i, \quad (y'_i \neq y_i \text{ 表示预测的标签不正确})$$

- 分类器权重 g_t

权重与错误率成反比, 表示其在投票时的重要性

$$g_t = \log \frac{1 - err(C_t)}{err(C_t)}$$

- 样本权重怎么调整?

- ✓ 所有样本权重之和始终为1, 且初始样本权重相等 $w_i = \frac{1}{n}$;
- ✓ 若当前分类器对该样本分类正确, 则用以下公式减小权重, 并重新归一化 $\sum_{i=1}^n w_i = 1$ 。

$$w_{i-new} = w_{i-old} * \frac{err(C_t)}{1 - err(C_t)}$$

Bagging vs Boosting

Bagging的特点： 独立训练分类器

关键： 对每个基分类器使用不同的训练数据

好处： 方便并行化和分布式数据处理

Boosting的特点： 依赖于前面分类器的判断，专注于“难分样本”

好处： 在分类准确率的提升能比bagging更高

缺点： 可能会过拟合、不容易并行化

预测阶段除了投票/加权投票，还有其他方法吗？

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    使用初级学习算法  $\mathcal{L}_t$ 
2:    $h_t = \mathcal{L}_t(D)$ ;        产生初级学习器  $h_t$ .
3: end for
4:  $D' = \emptyset$ ;
5: for  $i = 1, 2, \dots, m$  do
6:   for  $t = 1, 2, \dots, T$  do
7:      $z_{it} = h_t(\mathbf{x}_i)$ ;    生成次级训练集.
8:   end for
9:    $D' = D' \cup ((z_{i1}, z_{i2}, \dots, z_{iT}), y_i)$ ;
10: end for
11:  $h' = \mathcal{L}(D')$ ;
输出:  $H(\mathbf{x}) = h'(h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_T(\mathbf{x}))$ 
```

特点: 训练一个额外的分类器代替简单的投票。

把基分类器的预测结果看成新的数据, 用其学习一个次级分类器。

科研前沿

集成学习中，各个基分类器的训练要么完全独立（bagging），要么只有有限的依赖关系（boosting中前一个基分类器训练结束才去训练下一个）。能不能在每个基分类器训练的过程中就建立相互学习的机制呢？

最近，在深度学习的模型学习中，发表的相关论文采用online group learning的模式，对所有在group中的分类器同时训练，利用同伴当前的知识指导某一学习器的学习^[1]，或者对模型参数进行整合直接得到一个新的模型^[2]。

[1] [Defang Chen](#), [Jian-Ping Mei](#), [Can Wang](#), [Yan Feng](#), [Chun Chen](#):Online Knowledge Distillation with Diverse Peers online knowledge distillation with diverse peers, *AAAI 2020*.[pdf](#)[code](#)

[2] [Antti Tarvainen](#), [Harri Valpola](#):Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *NIPS 2017: 1195-1204* [pdf](#)[code](#)