



西安交通大学
XI'AN JIAOTONG UNIVERSITY

IAIR Est. 1986

Institute of
Artificial Intelligence
and Robotics



人工智能学院
College of Artificial Intelligence, XJTU

西安交通大学人工智能学院魏平编写。课程资料，请勿外传

《高级机器学习》第四章

集成学习 Ensemble learning

魏平

西安交通大学人工智能学院
人工智能与机器人研究所

西安交通大学人工智能学院魏平编写。课程资料，请勿外传

IAIR Est. 1986
Institute of
Artificial Intelligence
and Robotics



人工智能学院
College of Artificial Intelligence, XJTU

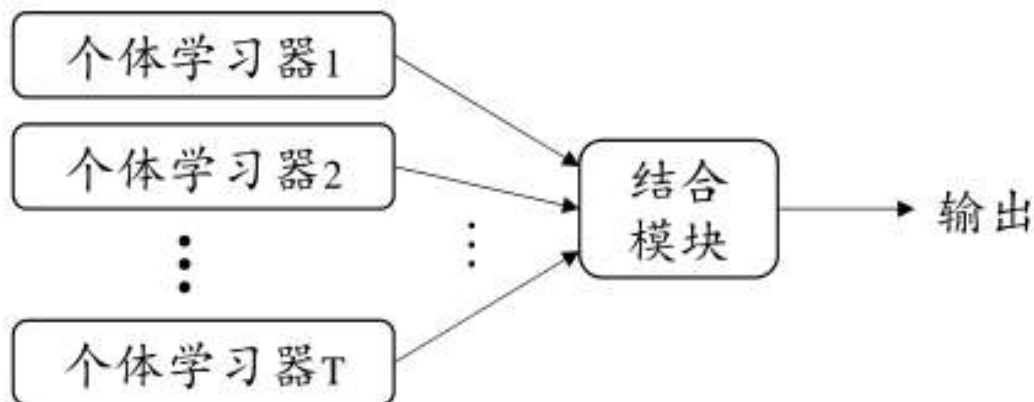
CONTENTS



- **个体与集成**
- **Boosting**
- **Bagging与随机森林**

个体与集成

- 集成学习(ensemble learning)通过构建并结合多个学习器来完成学习任务，提升性能；也称为多分类器系统(multi-classifier system)、基于委员会的学习(committee-based learning)



- 同质：集成中只包含同种类型的个体学习器，其中的个体学习器称为**基学习器**
- 异质：集成中包含不同类型的个体学习器，其中的个体学习器称为**组件学习器**

个体与集成

- 考虑一个简单的例子，在二分类问题中，假定3个分类器在三个样本中的表现如下图所示，其中√表示分类正确，X号表示分类错误，集成的结果通过投票产生

	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	×	×	√
集群	√	√	√	集群	√	√	×	集群	×	×	×
(a) 集群提升性能			(b) 集群不起作用			(c) 集群起负作用					

个体与集成

- 个体学习器应该“好而不同”
- 理论上可以证明，随着集成中个体分类器数目 T 的增大，集成的错误率将指数级下降，最终趋向于0
- 上述结论有一个假设：基学习器的误差相互独立。但个体学习器是为解决同一个问题训练出来的，显然不可能独立。因此，个体学习器的“准确性”和“多样性”存在冲突。如何产生并结合“好而不同”的个体学习器是集成学习的核心
- 集成学习分为两类：
 - 个体学习器间存在强依赖关系、必须串行生成的序列化方法，如Boosting
 - 个体学习器间不存在强依赖关系、可同时生成的并行化方法，如Bagging和随机森林 (Random Forest)

西安交通大学人工智能学院魏平编写。课程资料，请勿外传

IAIR Est. 1986
Institute of
Artificial Intelligence
and Robotics



人工智能学院
College of Artificial Intelligence, XJTU

CONTENTS

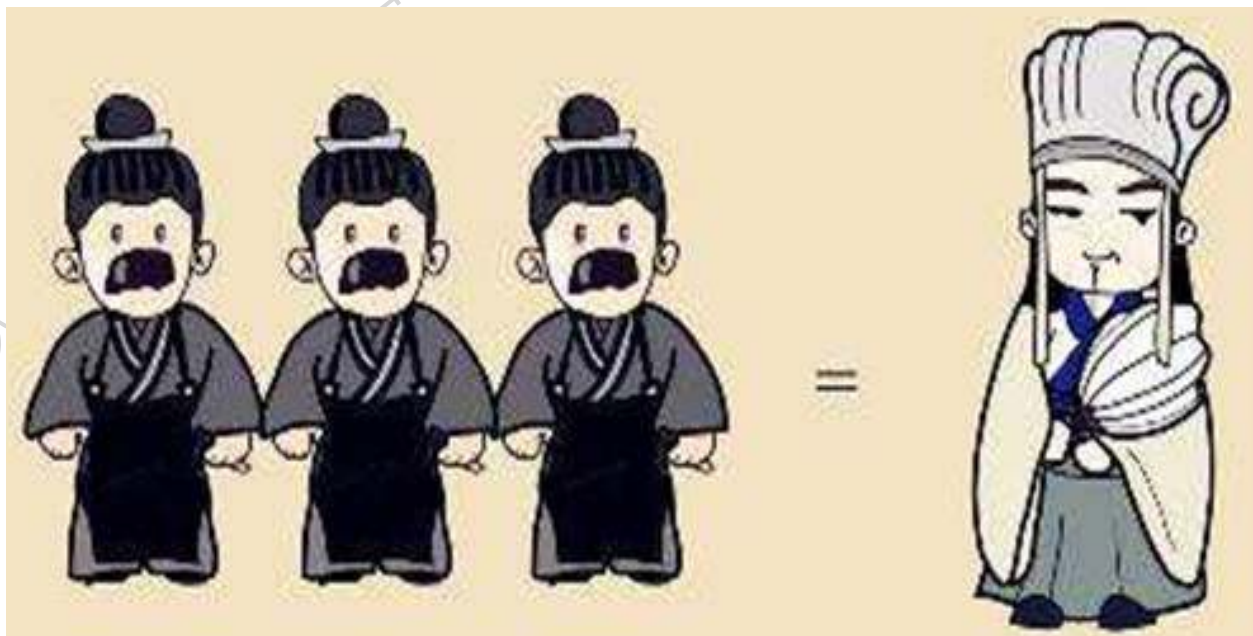


- 个体与集成
- **Boosting**
- **Bagging与随机森林**

Boosting

- ❑ Boosting是一族可将弱学习器提升为强学习器的算法
- ❑ Boosting基于这样一种思想：对于一个复杂任务来说，将多个专家的判断进行适当的综合所得的判断，要比其中任何一个专家单独的判断好

“三个臭皮匠顶个诸葛亮”

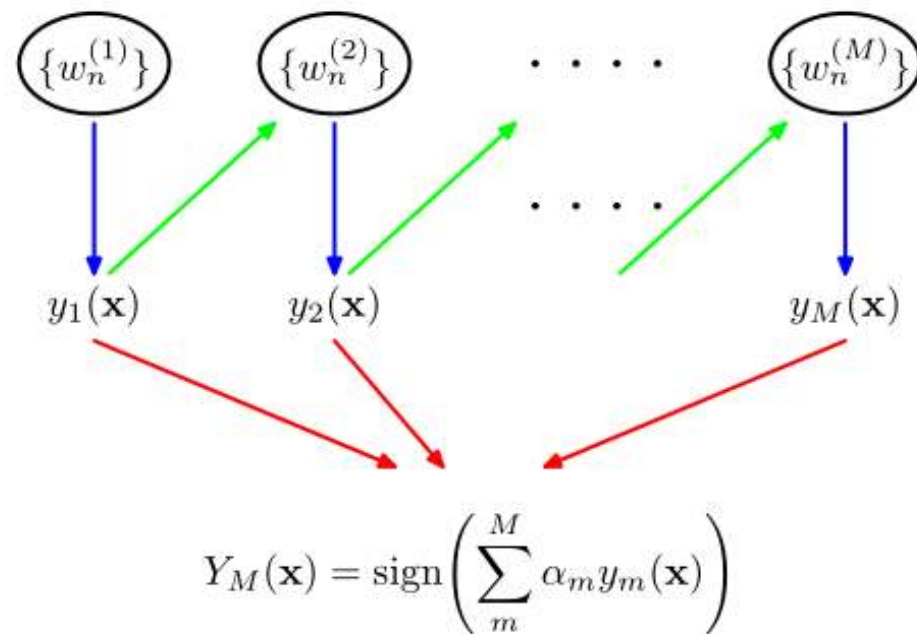


Boosting

□ Boosting机制

- ① 从初始训练集训练出一个基学习器
- ② 根据基学习器的表现对训练样本分布进行调整，使得先前基学习器做错的训练样本在后续受到更多关注
- ③ 基于调整后的样本分布来训练下一个基学习器

重复进行直到基学习器数目达到事先指定的值 T ，最终将这 T 个基学习器进行加权结合



Boosting

Input: Sample distribution \mathcal{D} ;
Base learning algorithm \mathcal{L} ;
Number of learning rounds T .

Process:

1. $\mathcal{D}_1 = \mathcal{D}$. % Initialize distribution
2. **for** $t = 1, \dots, T$:
3. $h_t = \mathcal{L}(\mathcal{D}_t)$; % Train a weak learner from distribution \mathcal{D}_t
4. $\epsilon_t = P_{\mathbf{x} \sim \mathcal{D}_t}(h_t(\mathbf{x}) \neq f(\mathbf{x}))$; % Evaluate the error of h_t
5. $\mathcal{D}_{t+1} = \text{Adjust_Distribution}(\mathcal{D}_t, \epsilon_t)$
6. **end**

Output: $H(\mathbf{x}) = \text{Combine_Outputs}(\{h_1(\mathbf{x}), \dots, h_t(\mathbf{x})\})$

□ Boosting算法特点

- 个体学习器存在强依赖关系，串行生成
- 每次调整训练数据的分布，使错分样本得到更多关注

□ Boosting算法最著名的代表是AdaBoost

AdaBoost算法 (Adaptive Boosting)

- 二分类问题, $y_i \in \{-1, +1\}$, $f(x)$ 为真实函数, $h_t(x)$ 为第 t 个分类器

$$\left. \begin{array}{l} h_1(x) \in \{-1, +1\} \\ h_2(x) \in \{-1, +1\} \\ \vdots \\ h_T(x) \in \{-1, +1\} \end{array} \right\} \begin{array}{l} H_T(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right) \\ \text{strong classifier} \end{array}$$

Weak classifiers

AdaBoost算法

□ 二分类问题, $y_i \in \{-1, +1\}$, $f(x)$ 为真实函数, $h_t(x)$ 为第 t 个分类器

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: $D_1(\mathbf{x}) = 1/m$. 初始化样本权值分布
 - 2: **for** $t = 1, 2, \dots, T$ **do**
 - 3: $h_t = \mathcal{L}(D, D_t)$; 基于分布 D_t 从数据集 D 中训练出分类器 $h_t(x)$
 - 4: $\epsilon_t = P_{\mathbf{x} \sim D_t}(h_t(\mathbf{x}) \neq f(\mathbf{x}))$; 估计 h_t 的误差
 - 5: **if** $\epsilon_t > 0.5$ **then break**
 - 6: $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$; 确定 h_t 的权重, 误分类误差率越小的基本分类器在最终分类器的作用越大
 - 7: $D_{t+1}(\mathbf{x}) = \frac{D_t(\mathbf{x})}{Z_t} \times \begin{cases} \exp(-\alpha_t), & \text{if } h_t(\mathbf{x}) = f(\mathbf{x}) \\ \exp(\alpha_t), & \text{if } h_t(\mathbf{x}) \neq f(\mathbf{x}) \end{cases}$ 更新样本分布 D_t, Z_t 为规范化因子, 确保 D_{t+1} 为一个分布, 误分类样本的权值得以扩大, 被正确分类的样本的权值得以缩小

$$= \frac{D_t(\mathbf{x}) \exp(-\alpha_t f(\mathbf{x}) h_t(\mathbf{x}))}{Z_t}$$
 - 8: **end for**
- 输出: $H(\mathbf{x}) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(\mathbf{x}) \right)$ 输出为基学习器的线性加权组合

例1：数字分类

- 给定如下所示的训练数据，假设弱分类器由 $x < v$ 或 $x > v$ 产生，其阈值 v 使得该分类器在数据集上分类误差最低。试用AdaBoost算法学习一个强分类器。

训练数据集

序号	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

- 解：初始化数据权值分布： $D_1 = (w_{11}, w_{12}, \dots, w_{110})$
对 $m=1$
 $w_{1i} = 0.1, \quad i = 1, 2, \dots, 10$

- ① 在权值分布为 D_1 的训练器上，阈值 v 取2.5时分类误差率最低，故基本分类器为

$$G_1(x) = \begin{cases} 1, & x < 2.5 \\ -1, & x > 2.5 \end{cases}$$

例1：数字分类

② $G_1(x)$ 在训练数据集上的误差率 $e_1 = P(G_1(x_i) \neq y_i) = 0.3$

③ 计算 $G_1(x)$ 的系数： $\alpha_1 = \frac{1}{2} \log \frac{1-e_1}{e_1} = 0.4236$

④ 更新训练数据的权值分布：

序号	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

Z_1

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

$$f_1(x) = 0.4236G_1(x)$$

分类器在 $\text{sign}[f_1(x)]$ 在训练数据集上有3个误分类点

例1：数字分类

对 $m=2$ ：

① 在权值分布为 D_2 的训练器上，阈值 v 取8.5时分类误差率最低，故基本分类器为

$$G_2(x) = \begin{cases} 1, & x < 8.5 \\ -1, & x > 8.5 \end{cases}$$

序号	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

④ 更新训练数据的权值分布：

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

$$f_2(x) = 0.4236G_1(x) + 0.6496G_2(x)$$

分类器在 $\text{sign}[f_2(x)]$ 在训练数据集上有3个误分类点

例1：数字分类

对 $m=3$ ：

① 在权值分布为 D_3 的训练器上，阈值 v 取5.5时分类误差率最低，故基本分类器为

$$G_3(x) = \begin{cases} 1, & x > 5.5 \\ -1, & x < 5.5 \end{cases}$$

② $G_3(x)$ 在训练数据集上的误差率 $e_3 = 0.1820$ 。

③ 计算 $G_2(x)$ 的系数： $\alpha_2 = 0.7514$

④ 更新训练数据的权值分布：

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

$$f_3(x) = 0.4236G_1(x) + 0.6496G_2(x) + 0.7514G_3(x)$$

分类器在 $\text{sign}[f_3(x)]$ 在训练数据集上有0个误分类点，得到最终分类器：

$$G(x) = \text{sign}[f_3(x)] = \text{sign}[0.4236G_1(x) + 0.6496G_2(x) + 0.7514G_3(x)]$$

例2：人脸检测

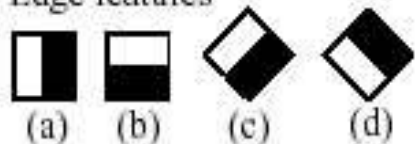
西安交通大学人工智能学院魏平编写。课程资料，请勿外传



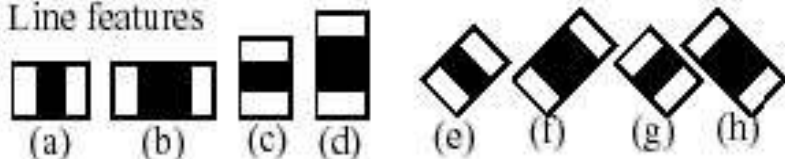
例2：人脸检测

西安交通大学人工智能学院魏平编写。课程资料，请勿外传

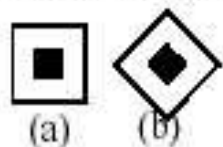
1. Edge features



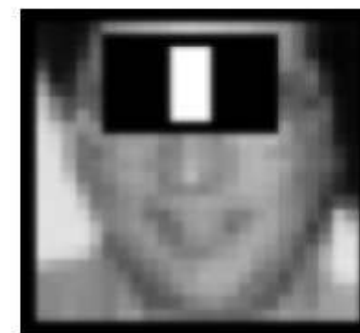
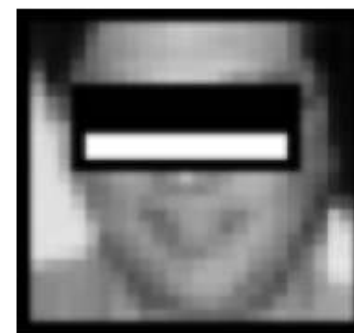
2. Line features



3. Center-surround features



Haar-like特征



Viola, P.A., Jones, M. Robust Real-time Object Detection. International Journal of Computer Vision, 2001

提取人脸区域图像特征

例2：人脸检测

西安交通大学人工智能学院魏平编写。课程资料，请勿外传

AdaBoost算法

- Given example images $(x_1, y_1), \dots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \dots, T$:

1. Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

so that w_t is a probability distribution.

2. For each feature, j , train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t , $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
3. Choose the classifier, h_t , with the lowest error ϵ_t .
4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

例2：人脸检测

西安交通大学人工智能学院魏平编写。课程资料，请勿外传

训练数据



检测结果





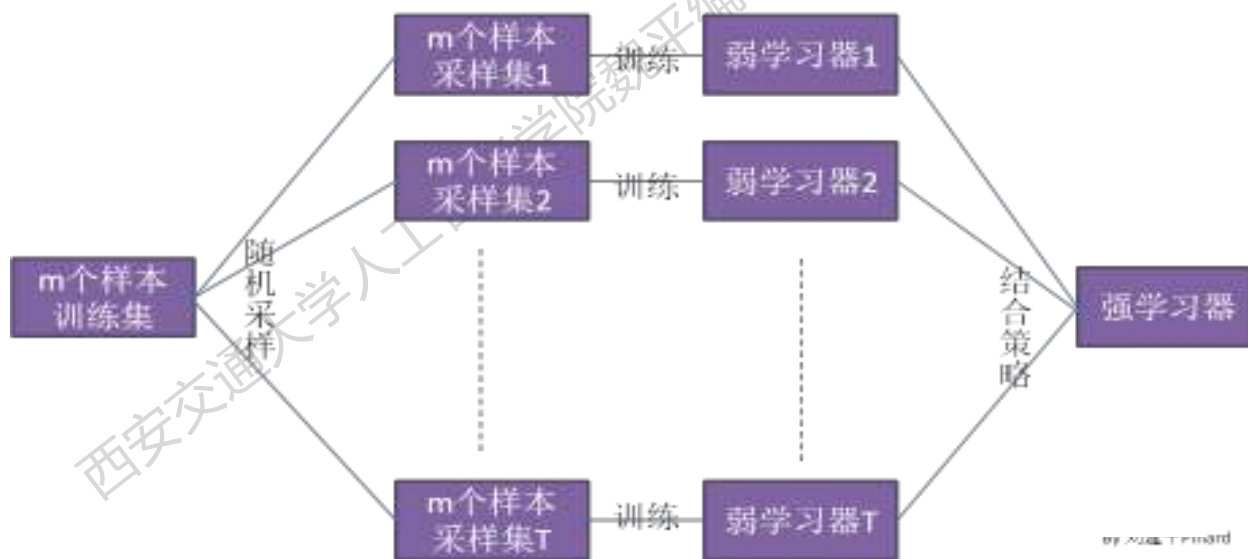
CONTENTS



- **个体与集成**
- **Boosting**
- **Bagging与随机森林**

Bagging方法

- 个体学习器不存在强依赖关系，并行生成
- 自助采样法bootstrap sampling: 给定 m 个样本的数据集，先随机取出一个样本放入采样集中，再把该样本放回初始数据集，使得下次采样时该样本仍有可能被选中。经过 m 次随机采样操作，得到 m 个样本集



Bagging方法

- Bagging算法基本流程：采样出 T 个含 m 个训练集的采样集，然后基于每个采样集训练出一个基学习器，再将这些学习器相结合
- 在预测输出进行结合时，对分类任务使用简单投票法，对回归任务使用简单平均法

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: **for** $t = 1, 2, \dots, T$ **do**
2: $h_t = \mathcal{L}(D, \mathcal{D}_{bs})$
3: **end for**

输出： $H(\mathbf{x}) = \arg \max_{y \in \mathcal{Y}} \sum_{t=1}^T \mathbb{I}(h_t(\mathbf{x}) = y)$

Bagging方法

- 输入为样本集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ 弱学习器算法, 弱分类器迭代次数 T
- 输出为最终的强分类器 $f(x)$
 - 对于 $t=1, 2, \dots, T$:
 - ① 对训练集进行第 t 次随机采样, 共采集 m 次, 得到包含 m 个样本的采样集 D_t
 - ② 用采样集 D_t 训练第 t 个弱学习器 $G_t(x)$
 - 如果是分类算法预测, 则 T 个弱学习器投出最多票数的类别或者类别之一为最终类别。如果是回归算法, T 个弱学习器得到的回归结果进行算术平均得到的值为最终的模型输出

Bagging算法特点：

□ 时间复杂度低

- 假定基学习器的计算复杂度为 $O(m)$ ，采样与投票/平均过程的复杂度为 $O(s)$ ，则 bagging 的复杂度大致为 $T(O(m)+O(s))$
- 由于 $O(s)$ 很小且 T 是一个不大的常数，因此训练一个 bagging 集成与直接使用基学习器的复杂度同阶（高效）

□ 可方便推广至多分类及回归任务

- 与标准的AdaBoost只适用于二分类任务不同，Bagging能不经修改的用于多分类、回归任务

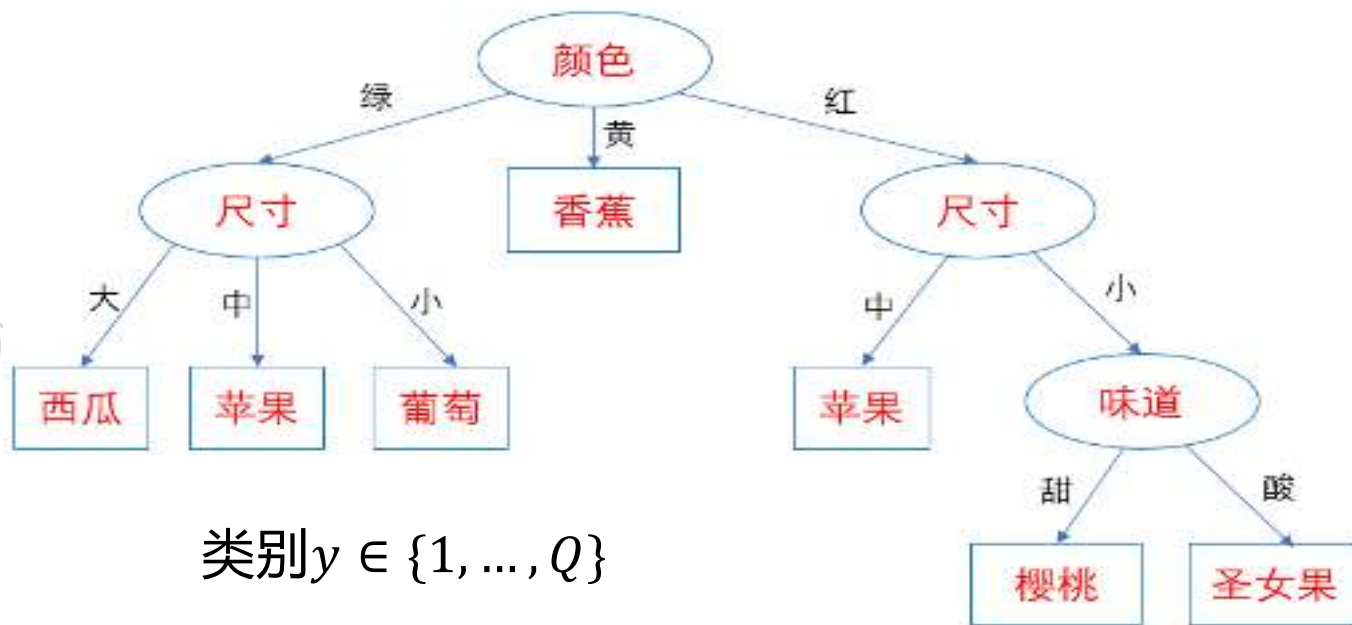
决策树(Decision Tree)

西安交通大学人工智能学院魏平编写。课程资料，请勿外传

- 决策树是一种树形结构，由节点和有向边组成，用于分类或回归。节点有两种：内部节点和叶节点，其中内部节点表示一个特征或属性，叶节点表示一个类，根节点表示开始测试的特征或属性；每个分支代表一个属性的判断结果输出



$$x = \{x^1, x^2, \dots, x^n\}^T$$



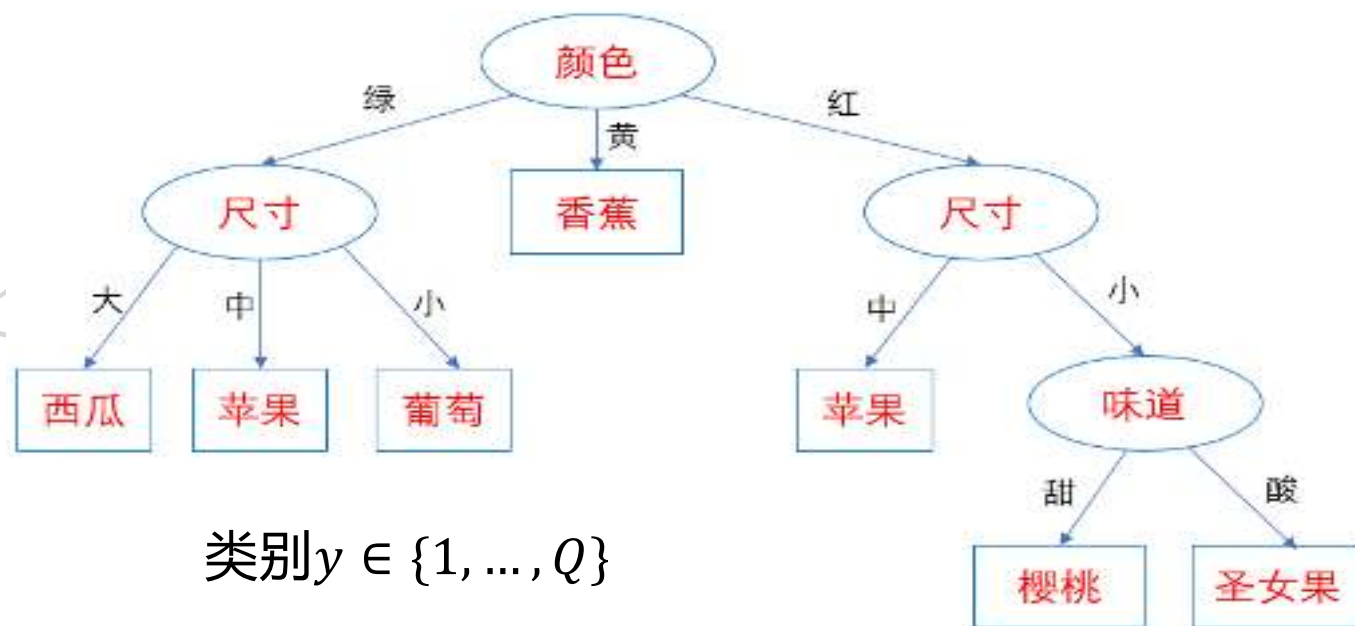
用决策树分类

- 分类树是一种对实例进行分类的决策树结构。在使用分类树进行分类时，从根节点开始，对实例的某一特征进行测试，根据测试结果，将实例分配到其子节点。每一个子节点对应着该特征的一个取值。如此递归地对实例进行测试并分配，直至达到叶节点，最后将实例分到叶节点的类中

输入数据

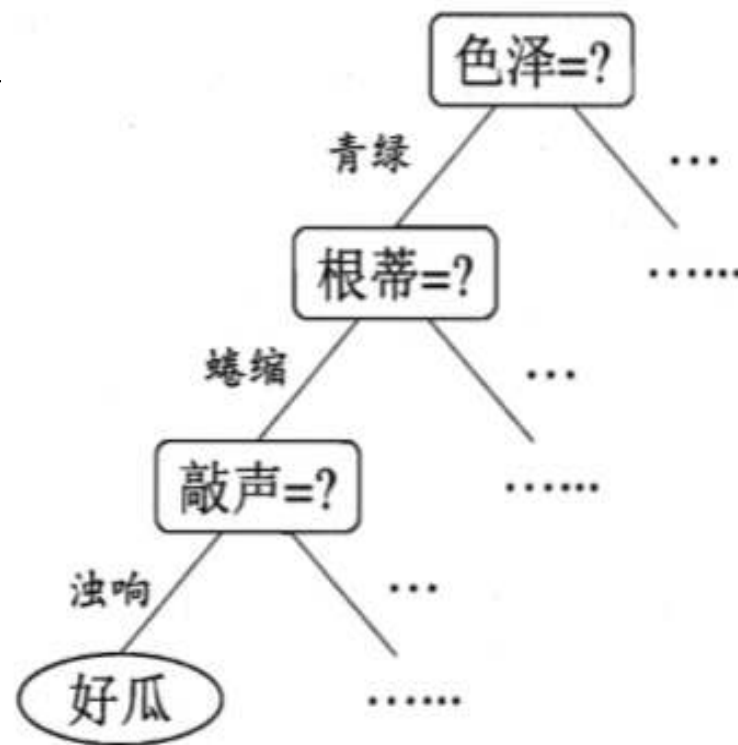


$$x = \{x^1, x^2, \dots, x^n\}^T$$



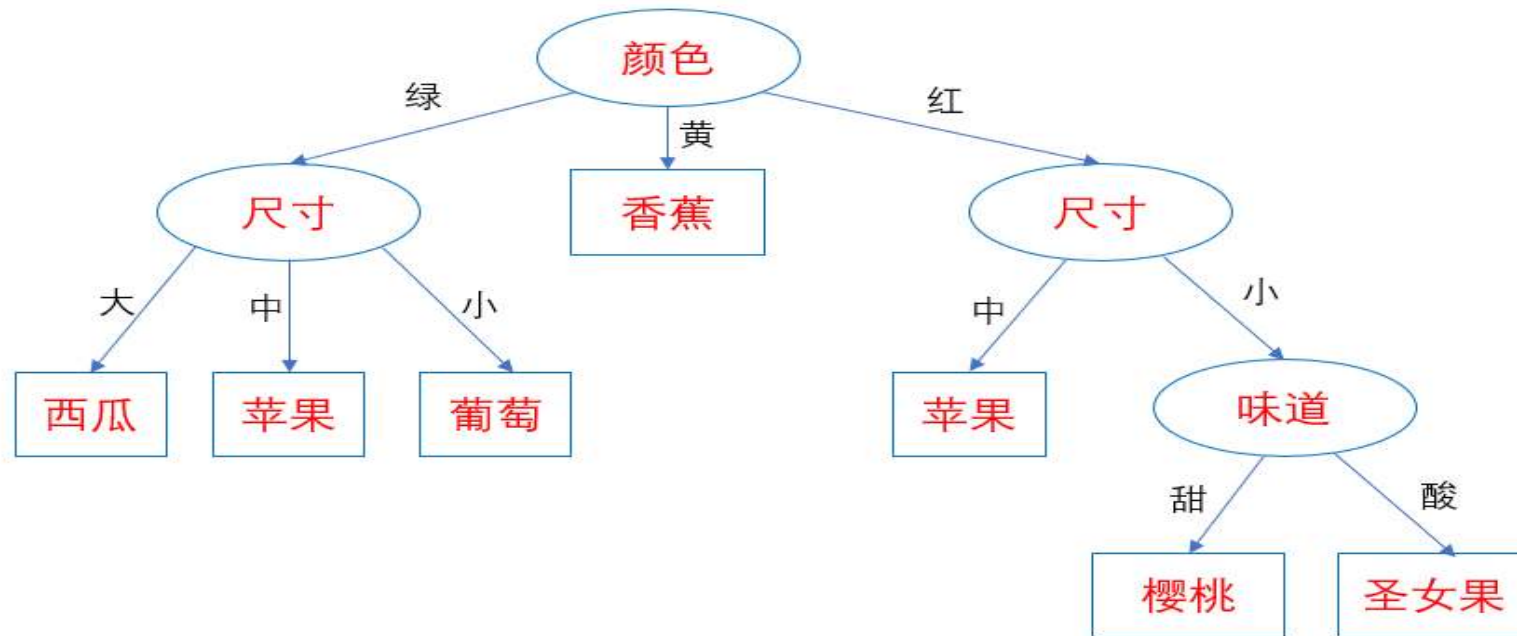
学习：构建分类树

- 分类树学习的目标是根据给定的训练数据集构建一个决策树模型，使它能够对实例进行正确的分类
- 给定训练数据集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$
 - $x_i = \{x_i^1, x_i^2, \dots, x_i^n\}^T$ 为输入数据， n 为特征维度
 - $y_i \in \{1, \dots, Q\}$ 为目标类别
- 学习目标：构造一个决策树结构及参数



学习：构建分类树

- 决策树学习的算法通常是递归选择最优特征，并根据该特征对训练数据进行分割，使得各个子数据集有一个最好的分类。过程对应着对特征空间的划分，也对应着决策树的构建



输入:

训练集: $D = (x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$

属性集: $A = a_1, a_2, \dots, a_n$

过程:

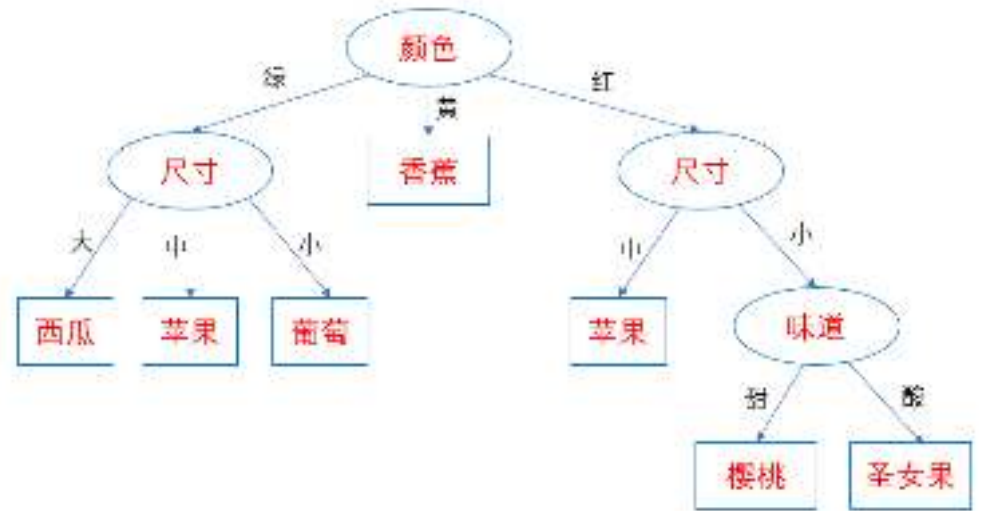
函数 $TreeGenerate(D, A)$

输出:

以node为根节点的决策树

算法:

- (1) 生成结点根node
- (2) **if** D 中样本全属于同一类别 C_k **then**
- (3) 将node标记为 C_k 类叶结点
- (4) **return**
- (5) **end if**
- (6) **if** $A = \emptyset$ **OR** D 中样本在 A 上取值相同 **then**
- (7) 将node标记为叶结点, 其类别标记为 D 中样本数最多的类
- (8) **return**
- (9) **end if**
- (10) 从 A 中选择最优划分属性 a_*
- (11) **for** a_* 的每一个值 a_*^v **do**
- (12) 为node生成一个分支: 令 D_v 表示 D 中在 a_* 上取值为 a_*^v 的样本子集
- (13) **if** D_v 为空 **then**
- (14) 将分支结点标记为叶结点, 其类别标记为 D 中样本最多的类
- (15) **return**
- (16) **else**
- (17) 以 $TreeGenerate(D_v, A - \{a_*\})$ 为分支结点
- (18) **end if**
- (19) **end for**



□ 算法流程

- 构建根节点, 将所有训练数据放在根节点, 选择一个最优特征, 按照这一特征将训练数据集分割成子集, 使得各个子集在当前条件下最好的分类
- 如果这些子集已经能够被基本正确分类, 那么构建叶节点, 并将这些子集分到所对应的叶节点中去
- 如果还有子集不能被基本正确分类, 那么就对这些子集选择新的最优特征, 继续对其进行分割, 构建相应的节点
- 如此递归地进行下去, 直至所有训练数据子集被基本正确分类, 或者没有合适的特征为止

构建分类树：如何选择最优划分属性

西安交通大学人工智能学院魏平编写。课程资料，请勿外传

- 从A中选择最优划分属性 a^* - 特征选择
- 如果利用一个特征进行分类的结果与随机分类的结果没有很大差别，则这个特征是没有分类能力的

区分樱桃和西红柿，“尺寸”特征比“颜色”特征明显，因此希望“尺寸”属性在决策树的上部

- 随着划分过程不断进行，我们希望决策树的分支节点所包含的样本尽可能属于同一类别，即节点的“纯度” (purity) 越来越高

如何度量分类纯度？

构建分类树：信息增益方法

西安交通大学人工智能学院魏平编写。课程资料，请勿外传

□ 信息熵 (information entropy)

设当前样本集合 D 中第 k 类样本所占的比例为 p_k ($k = 1, 2, \dots, Q$), 则 D 的信息熵定义为:

$$\text{Ent}(D) = - \sum_{k=1}^Q p_k \log_2 p_k$$

$\text{Ent}(D)$ 的值越小, 则 D 的纯度越高

□ 设离散属性 a 有 V 个可能的取值 $\{a^1, a^2, \dots, a^V\}$, 若使用 a 来对样本集进行划分, 则会产生 V 个分支节点, 其中第 v 个分支节点包含了 D 中所有在属性 a 上取值为 a^v 的样本, 记为 D^v 。用属性 a 对样本集 D 进行划分所得到的信息增益 (information gain) 为:

$$\text{Gain}(D, a) = \text{Ent}(D) - \sum_{v=1}^V \frac{|D^v|}{|D|} \text{Ent}(D^v)$$

则最优属性为: $a^* = \operatorname{argmax}_{a \in A} \text{Gain}(D, a)$

随机森林 (Random Forest)

- ❑ 随机森林(Random Forest, 简称RF)是bagging的一个扩展变种。RF在以决策树为基学习器构建Bagging集成的基础上，进一步在决策树的训练过程中引入了随机属性选择
- ❑ 具体来说：传统的决策树在选择划分属性时，在当前节点的属性集合（假定有 n 个属性）中选择一个最优属性；在RF中：对基决策树的每个节点，先从该节点的属性集合中随机选择一个包含 K 个属性的子集，然后再从这个子集中选择一个最优属性用于划分
- ❑ 参数 K 控制了随机性的引入程度：若令 $K = n$ ，则基决策树的构建与传统决策树相同；若 $K = 1$ ，则是随机选择一个属性用于划分；一般情况下，推荐值 $K = \log_2 n$

随机森林算法

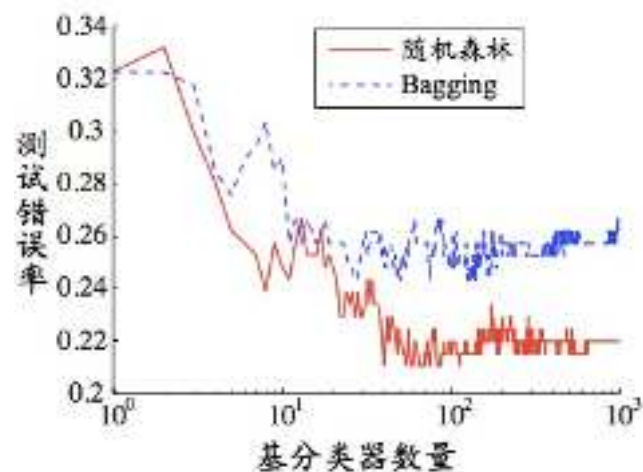
- 输入为样本集 $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ 弱学习器算法, 弱分类器迭代次数 T
- 输出为最终的强分类器 $f(x)$
 - 对于 $t = 1, 2, \dots, T$:
 - ① 对训练集进行第 t 次随机采样, 共采集 m 次, 得到包含 m 个样本的采样集 D_t
 - ② 用采样集 D_t 训练第 t 个决策树模型 $G_t(x)$ 。在训练决策树模型的节点的时候, 在节点上所有的样本特征中选择 K 个样本特征, 在这些随机选择的部分样本特征中选择一个最优的特征来做决策树的左右子树划分
 - 如果是分类算法预测, 则 T 个弱学习器 $G_t(x) (t = 1, 2, \dots, T)$ 投出最多票数的类别或者类别之一为最终类别

随机森林

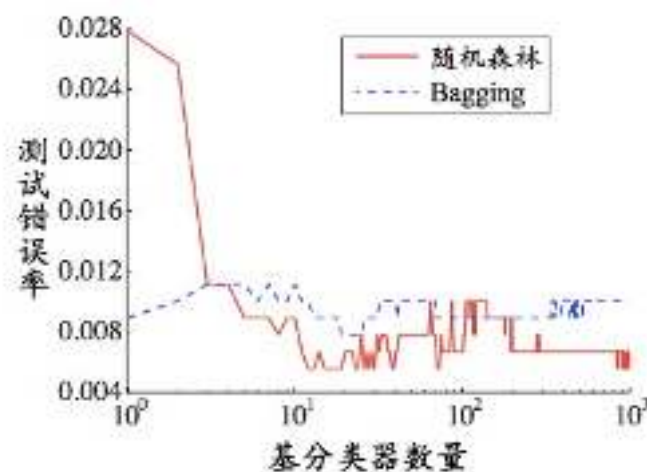
- 随机森林简单、容易实现、计算开销小，在许多现实任务中展现出强大的性能——“代表集成学习技术水平的方法”
- RF只对Bagging做了小改动：Bagging中基学习器的多样性仅通过**样本扰动**（通过对初始训练集采样）而来；RF中基学习器的多样性不仅来自**样本扰动**，还来自**属性扰动**。这使得最终集成的**泛化性能**可通过个体学习器之间差异度的增加而进一步提升
- 随机森林的训练效率通常优于Bagging，因为在个体决策树的构建过程中，Bagging使用的是“确定型”决策树，在选择划分属性时要对节点的所有属性进行考察，而RF使用的“随机型”决策树只需考察一个属性子集

随机森林实验

- 随机森林的收敛性与Bagging相似，如图所示，随机森林的起始性能往往相对较差，特别是在集成中只包含一个基学习器时，因为通过引入属性扰动，随机森林中个体学习器的性能往往有所降低，然而随着个体学习器数目的增加，随机森林通常会收敛到更低的泛化误差



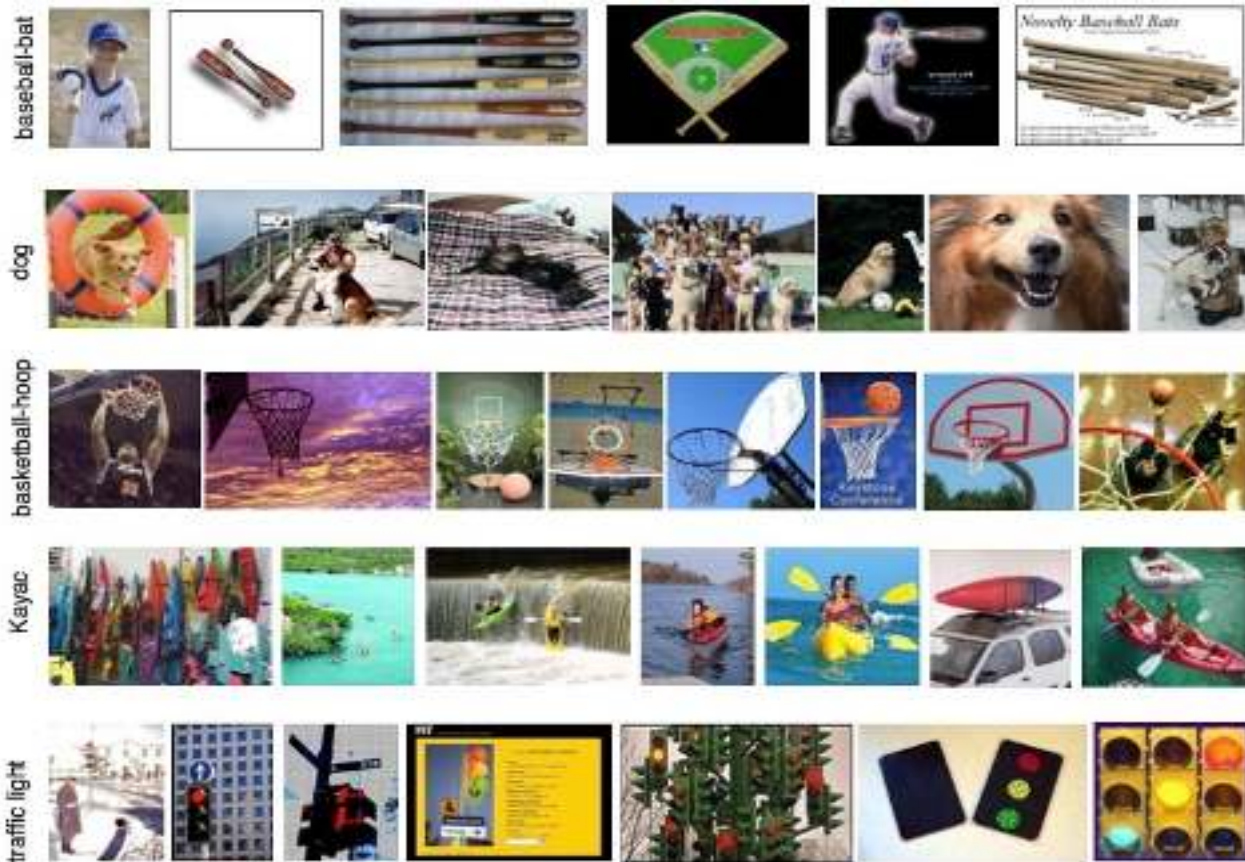
(a) glass 数据集



(b) auto-mpg 数据集

在两个UCI数据集上，集成规模对随机森林与Bagging的影响

例：图像分类 (Image Classification)

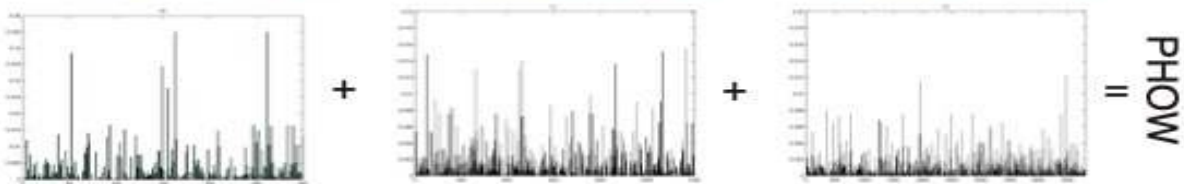


What is the class of an image?

A. Bosch, A. Zisserman and X. Munoz, "Image Classification using Random Forests and Ferns," *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1-8,

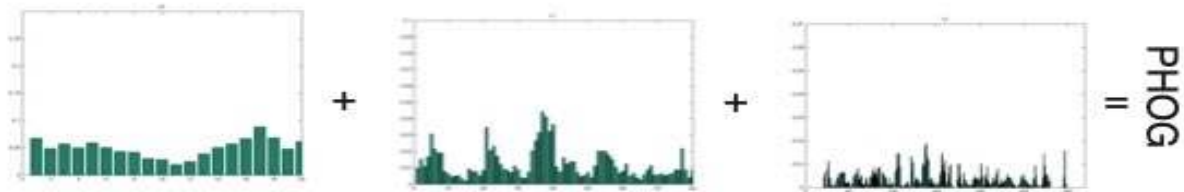
例：图像分类 (Image Classification)

□ 图像特征提取



PHOW

Pyramid Histogram Of visual Words (PHOW)



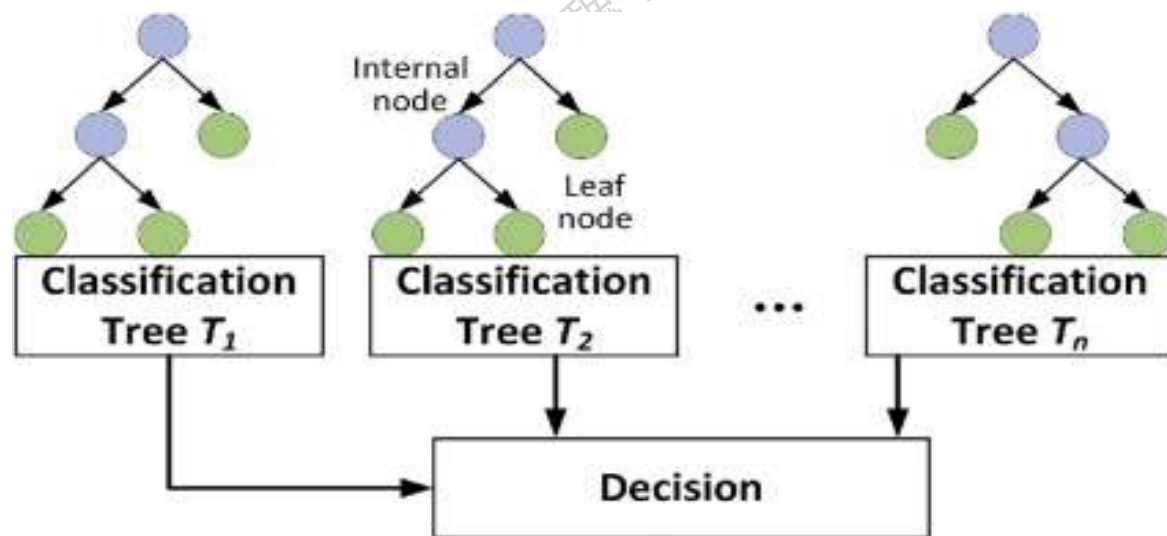
PHOG

Pyramid HOG

例：图像分类 (Image Classification)

随机森林构造与分类

- 每个叶子节点是类别的后验概率估计
- 每个内部节点是一个属性测试，将数据集分割
- 一个图像被送往每个树，最终结果是叶子节点分布的综合
- 随机性插入：1) 数据集子集采样；2) 节点测试选择



阅读材料

西安交通大学人工智能学院魏平编写。课程资料，请勿外传

- Zhi-Hua Zhou, Ensemble Methods: Foundations and Algorithms, Chapman and Hall/CRC; 1st edition, 2012
- Robert E. Schapire Yoav Freund, Boosting Foundations and Algorithms, The MIT Press, 2012

西安交通大学人工智能学院魏平编写。课程资料，请勿外传