Python
机器学习实战

随机森林和AdaBoost

随机森林和Adaboost

弱分类器和强分类器

怎样实现弱学习转为强学习

核心思想：通过组合使弱学习互补

随机森林和Adaboost

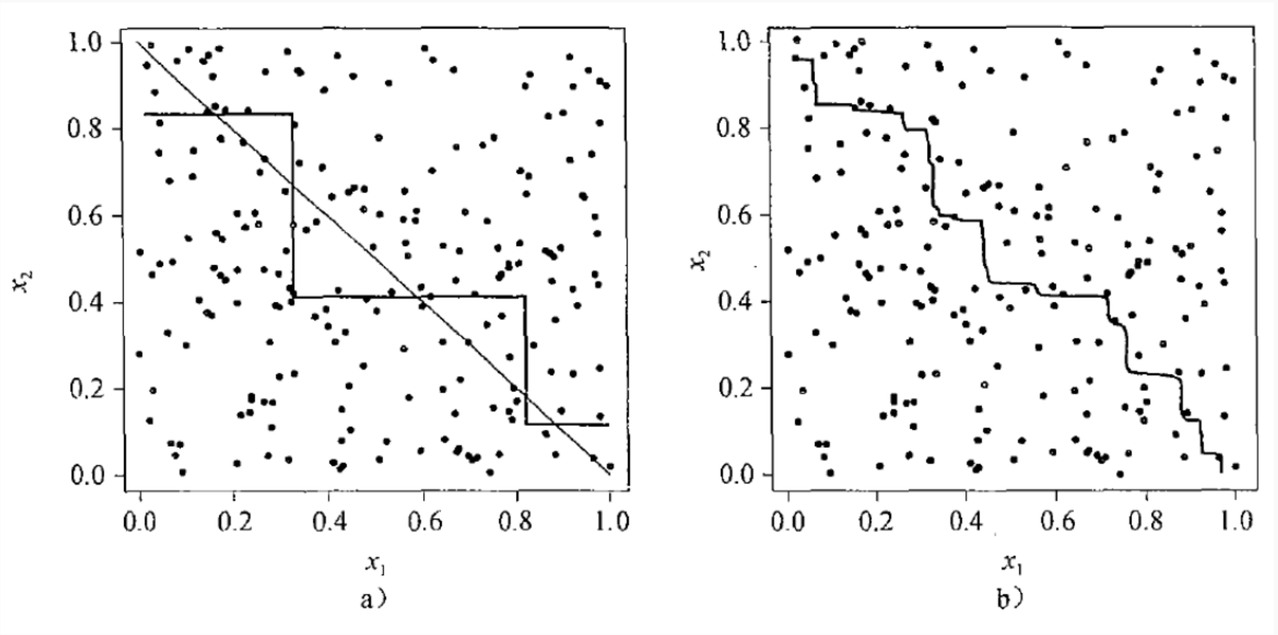
组合算法包括

装袋（bagging）

提升（boosting），Adaboost

随机森林

为啥组合算法那么NB



自助式抽样

从总体（m）中，有放回的方式抽取m个样本，组成的就是自助式抽样

$$p = \left(1 - \frac{1}{n}\right)^n$$

$$\lim_{n\to\infty} p = \lim_{n\to\infty} \left(1 - \frac{1}{n}\right)^n = e^{-1} \approx 0.368$$

装袋算法

Bagging（Bootstrap aggregating 的缩写）算法是最早的集成学习算法，具体的步骤可以描述为：

（1） 利用 Bootstrap 方法重采样，随机产生 T 个训练集 $S_1, S_2, \ldots, S_T$；

（2） 利用每个训练集，生成对应的决策树 $C_1, C_2, \ldots, C_T$；

（3） 对于测试集样本 X，利用每个决策树进行测试，得到对应的类别 $C_1(X), C_2(X), \ldots, C_T(X)$；

（4） 采用投票的方法，将 T 个决策树中输出最多的类别作为测试集样本 X 所属的类别。

随机森林和Adaboost

随机森林

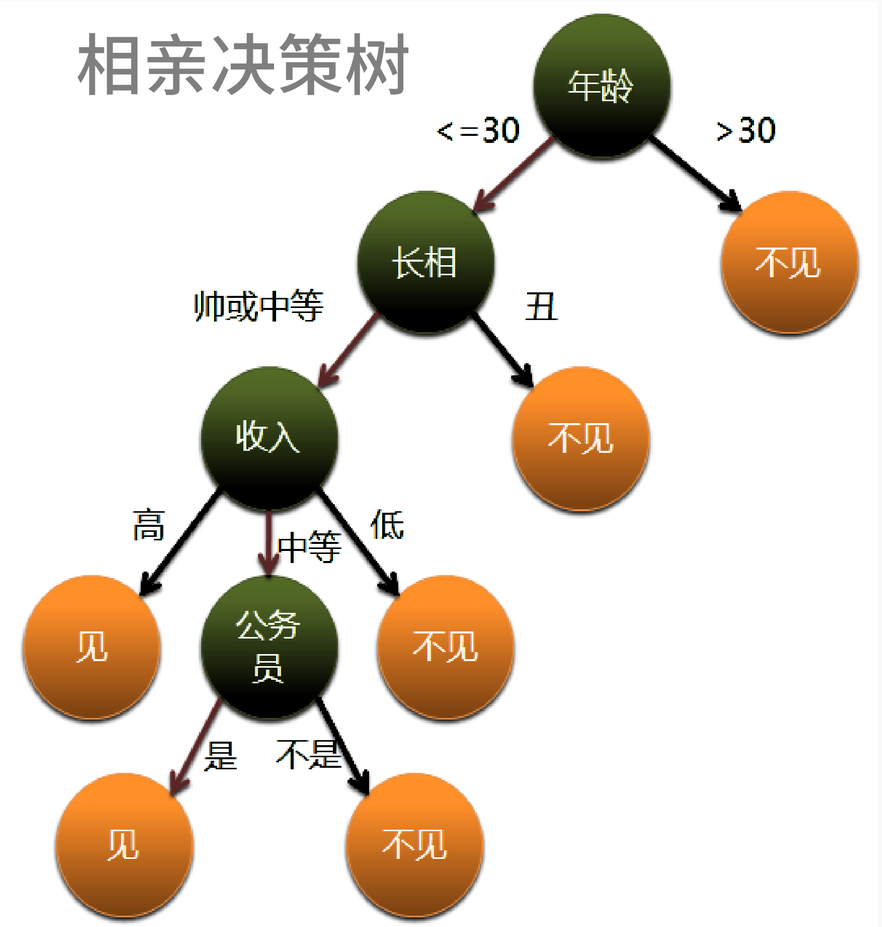
- 数据随机
- 属性随机

AdaBoost算法

决策树不需要剪枝

剪枝是为了避免过拟合

弱决策树不会过拟合



随机森林和Adaboost

Adaboost

是一种迭代算法，把弱分类器集合成一个更强的分类器

根据每次训练集中每个样本的分类是否正确，以及上次的总体分类的准确率，来确定每个样本的权值（即改变数据分布）。将修改过权值的新数据集送给下层分类器进行训练，最后将每次得到的分类器最后融合起来，作为最后的决策分类器

随机森林 & Adaboost

普罗大众 & 精英

Adaboost思想

1、先通过对N个训练样本的学习得到第一个弱分类器；

2、根据本次分类结果制造一个新的N个的训练样本，通过对这个样本的学习得到第二个弱分类器；

3、根据1、2的分类结果制造一个新的N个的训练样本，通过对这个样本的学习得到第三个弱分类器

4、最终经过提升的强分类器。即某个数据被分为哪一类要通过……的多数表决。

随机森林和Adaboost

流程

循环迭代，直到累积错误率为0{

① 更新样本分布D

② 获得当前分布下的最好弱分类器

③ 计算最好弱分类器的误差率ε

④ 计算最好弱分类器的话语权α

}

随机森林和Adaboost

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i] .$$

- Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$.
- Update:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

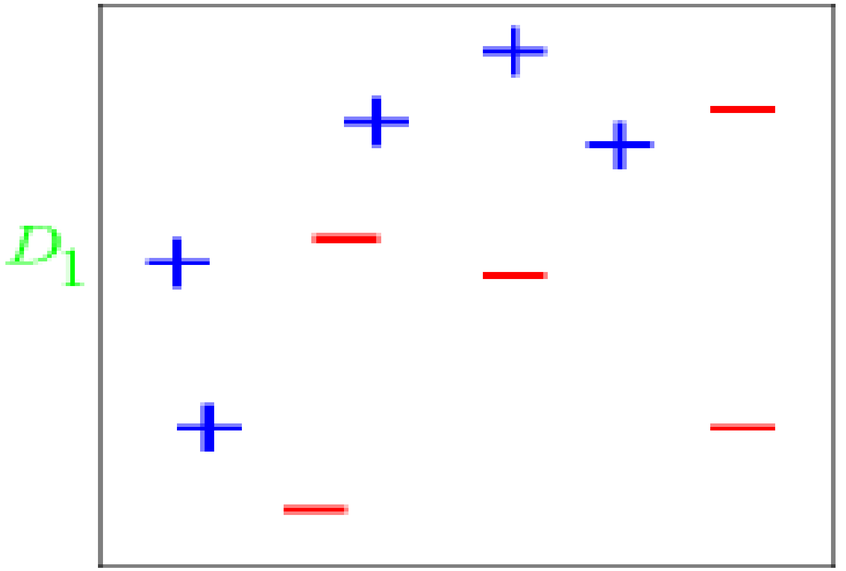
Output the final hypothesis:

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

Example

※ 样本：分类标签（1，-1）

※ 误差：正确划分误差为0，否则为1



※ “+”和“-”分别表示两种类别

※ 水平或者垂直的直线作为分类器

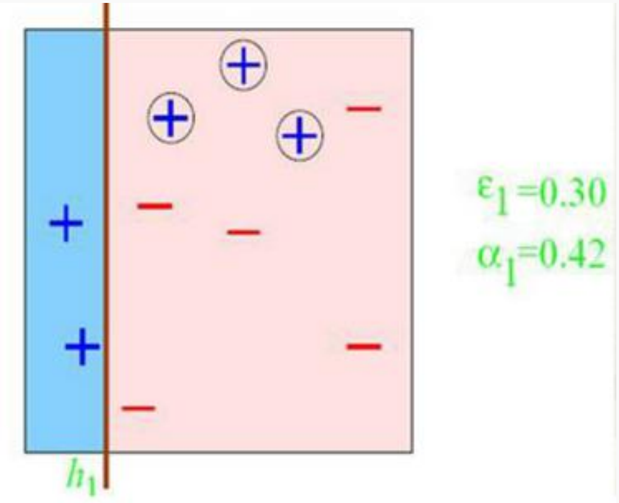
随机森林和Adaboost

第一轮

最开始均匀分布 D，所以h1 里的每个点的权值是0.1

误差为分错的三个点的值之和 $\varepsilon_1=(0.1+0.1+0.1)=0.3$

$$\alpha_1 = \frac{1}{2}\ln\left(\frac{1-\varepsilon_1}{\varepsilon_1}\right) = \frac{1}{2}\ln\left(\frac{1-0.3}{0.3}\right) = 0.42$$

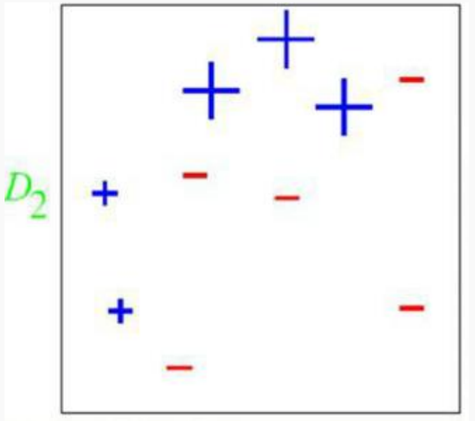


把分错点的权值变大：

对于分类正确的7个点，其权值保持不变，为0.1；

对于分类错误的3个点，其权值为

$$D_1(i)\frac{1-\varepsilon_1}{\varepsilon_1} = 0.1\left(\frac{1-0.3}{0.3}\right) = 0.2333$$



第二轮

分类错误的权值为：we2=0.1*3=0.3

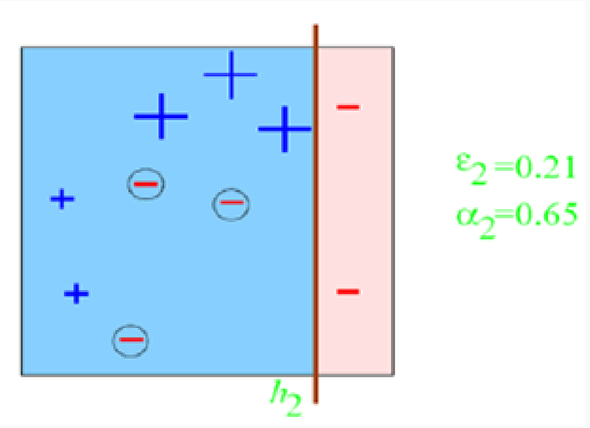
十个点的总权值为：wt2=0.1*7+0.233*3=1.3990

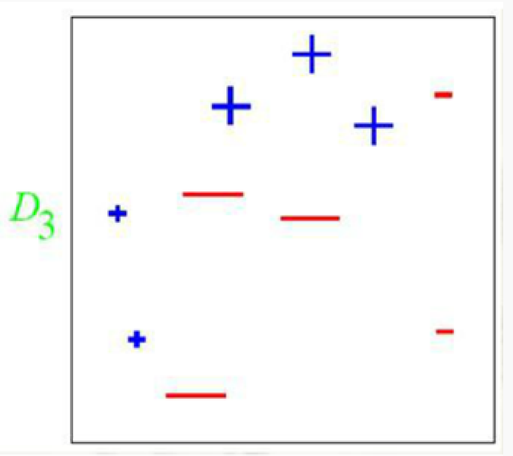
错误率为：ε2=we2/wt2=0.3/1.399= 0.2144

$$\alpha_2 = \frac{1}{2}\ln\left(\frac{1-\varepsilon_2}{\varepsilon_2}\right) = \frac{1}{2}\ln\left(\frac{1-0.2144}{0.2144}\right) = 0.6493$$

分类错误的三个点，其权值为:

$$D_2(\mathrm{i})\frac{1-\varepsilon_2}{\varepsilon_2} = 0.1\left(\frac{1-0.2144}{0.2144}\right) = 0.3644$$



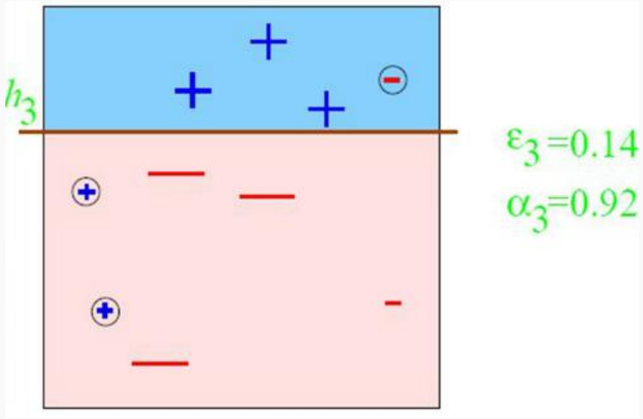


第三轮

分类错误的权值为we3=0.1*2+0.1*1=0.3

十个点的总权值为：wt_3=0.1*4+0.233*3+0.3664*3=2.1982

错误率为： $\varepsilon_t=we_3/wt_3$=0.3/2.1982= 0.1365



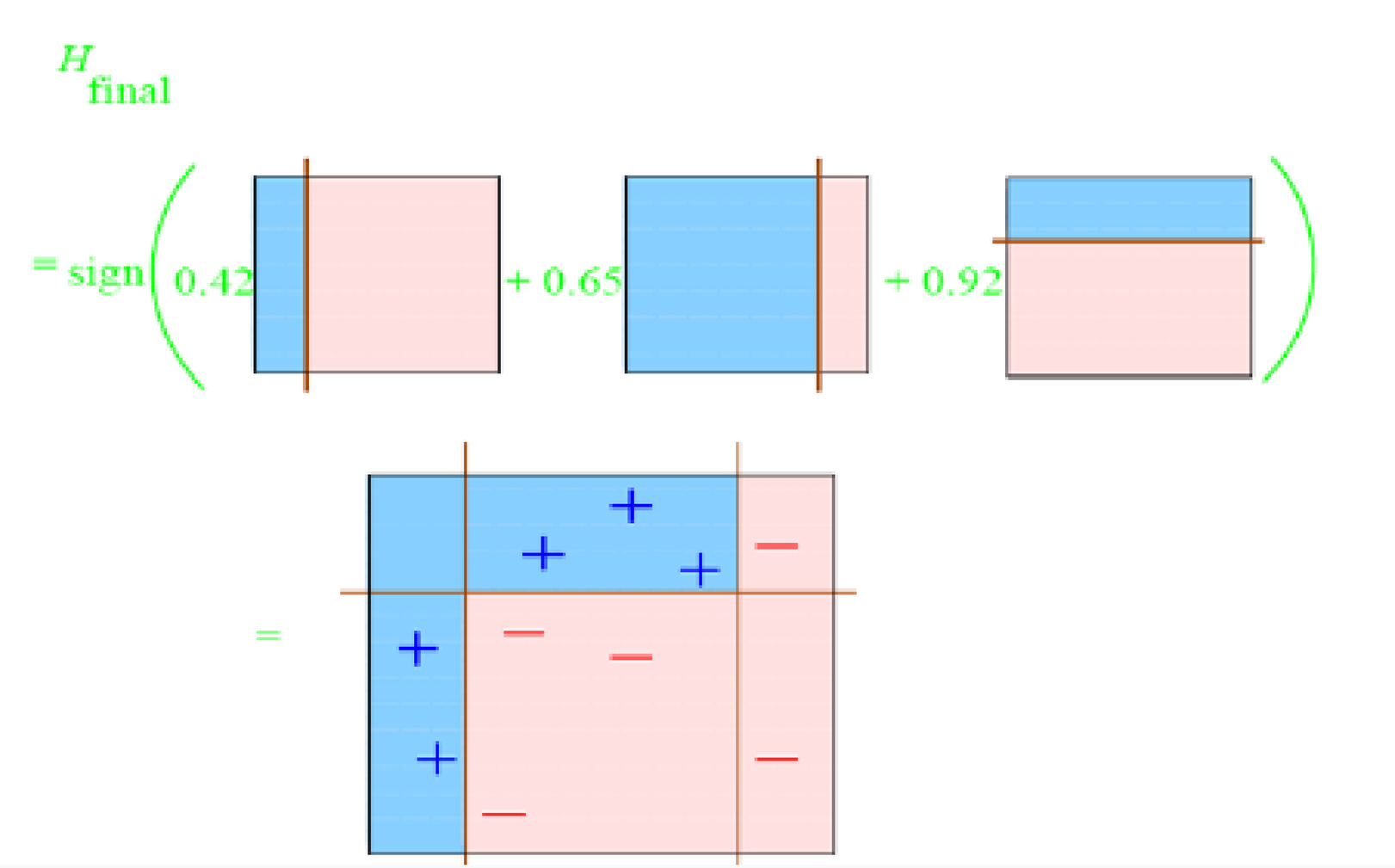
$$\alpha_3 = \frac{1}{2}\ln\left(\frac{1-\varepsilon_3}{\varepsilon_3}\right) = \frac{1}{2}\ln\left(\frac{1-0.1365}{0.1365}\right) = 0.9223$$

分类错误的三个点，其权值为:

$$D_3(\mathrm{i})\frac{1-\varepsilon_3}{\varepsilon_3} = 0.1\left(\frac{1-0.1365}{0.1365}\right) = 0.6326$$

随机森林和Adaboost

循环结束



实战

随机森林和Adaboost

Adaboost

弱分类器

强分类器

应用

人脸识别

数学推导