法律声明

- 本课件包括演示文稿、示例、代码、题库、视频和声音等内容，北风网和讲师拥有完全知识产权；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或者机构不得盗版、复制、仿造其中的创意和内容，我们保留一切通过法律手段追究违反者的权利。
- 课程详情请咨询
  - 微信公众号：北风教育
  - 官方网址：http://www.ibeifeng.com/

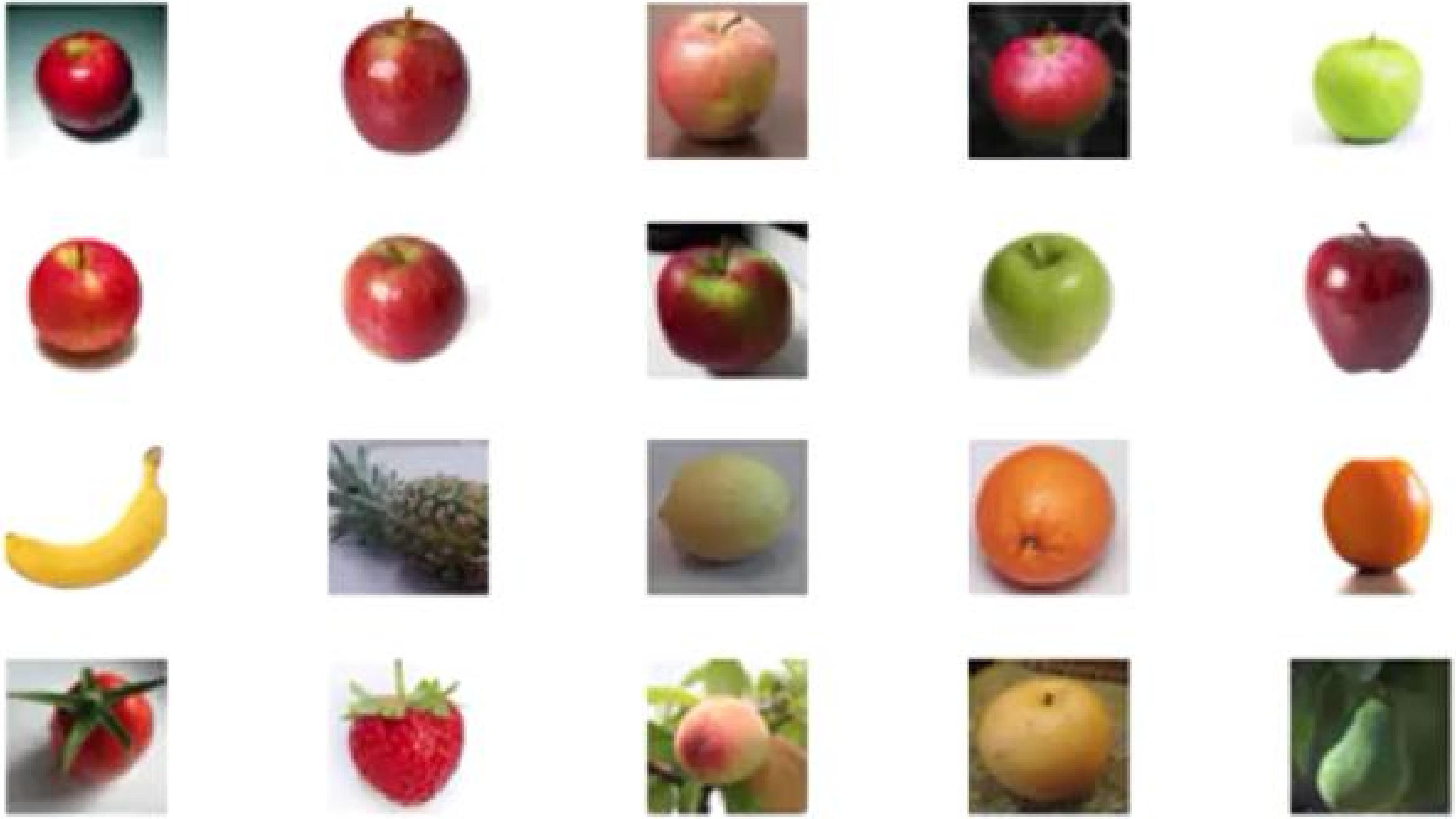
人工智能之机器学习

自适应提升算法（AdaBoost)

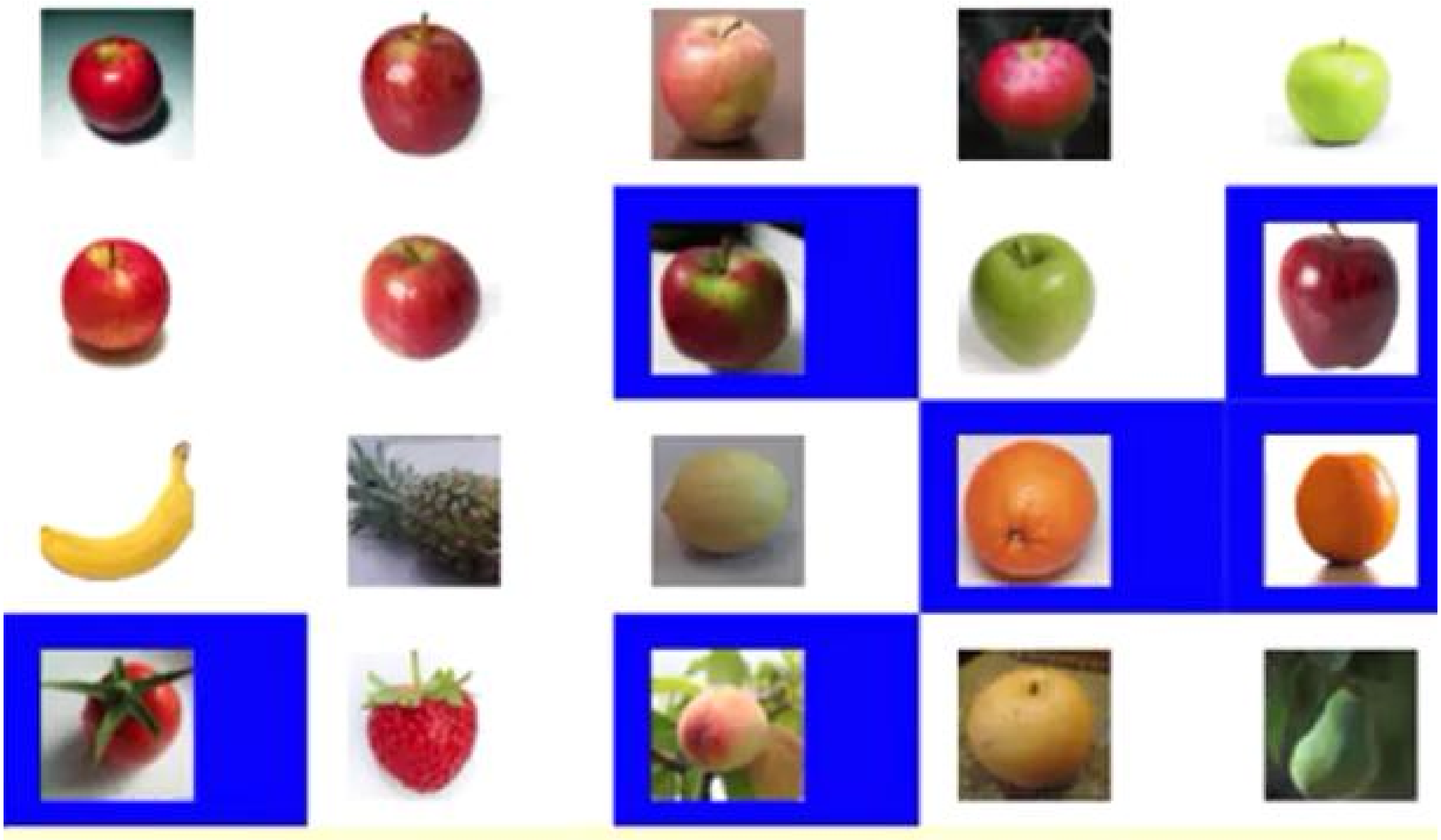
主讲人：赵翌臣

上海育创网络科技有限公司

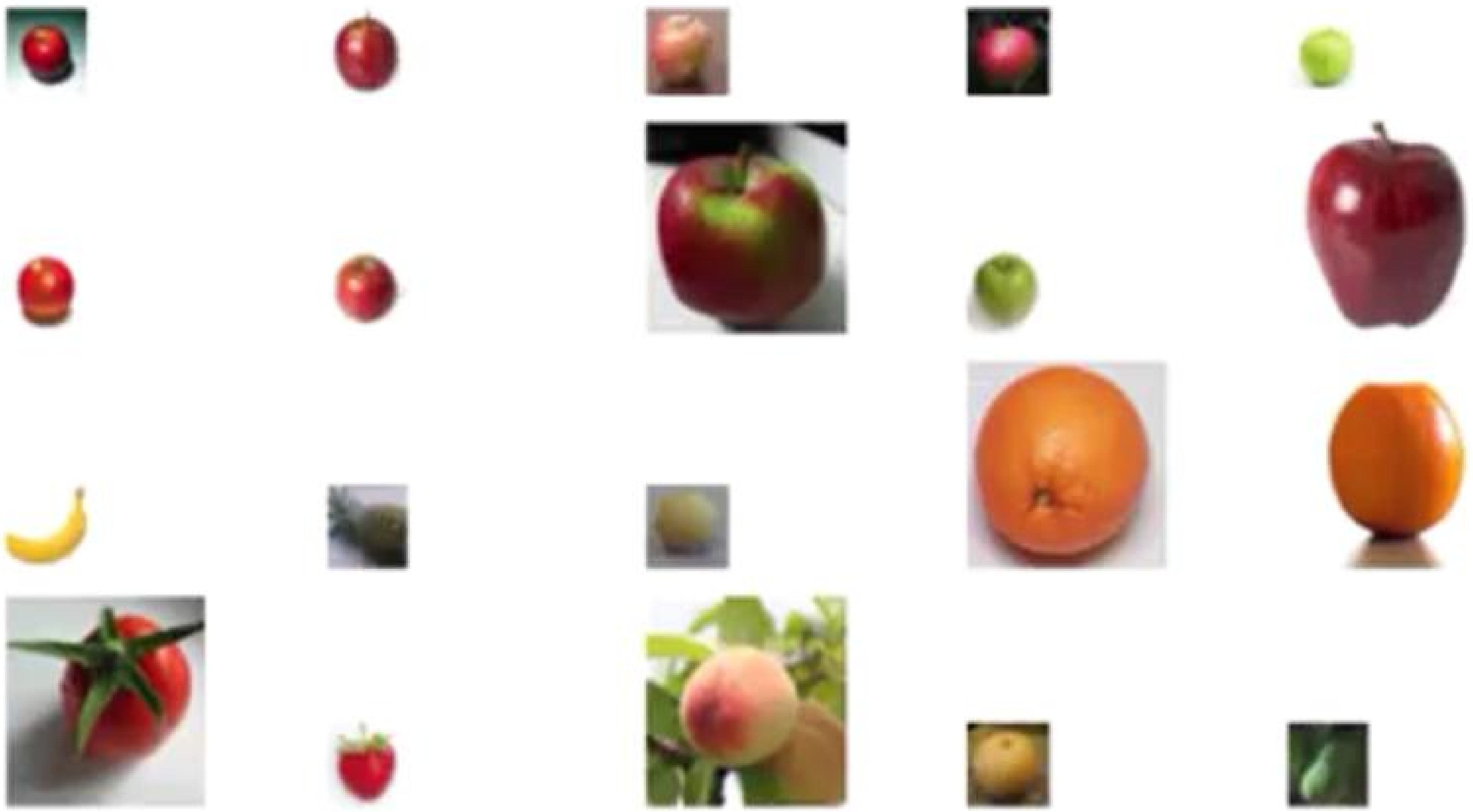
AdaBoost直观理解



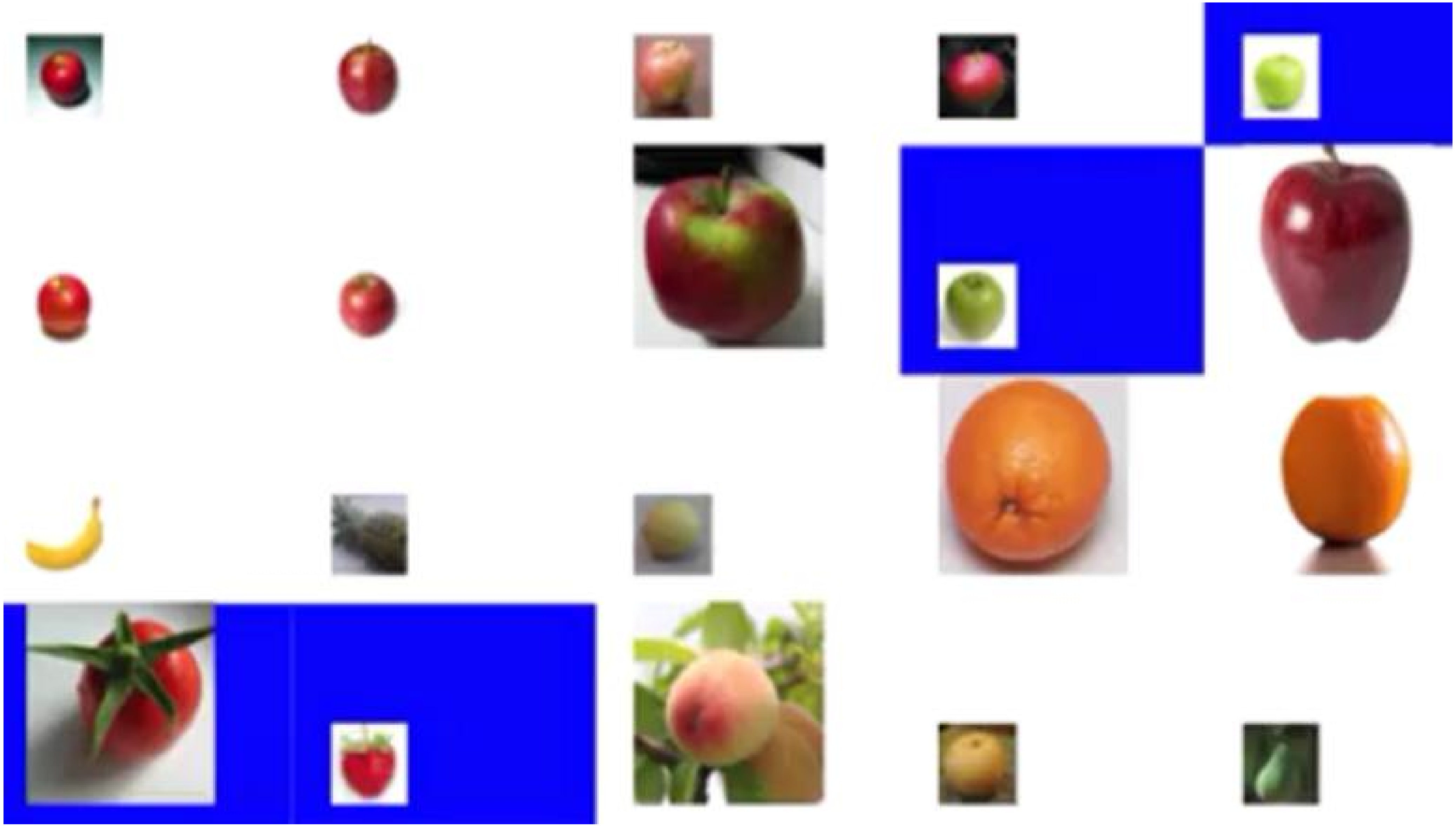
AdaBoost直观理解



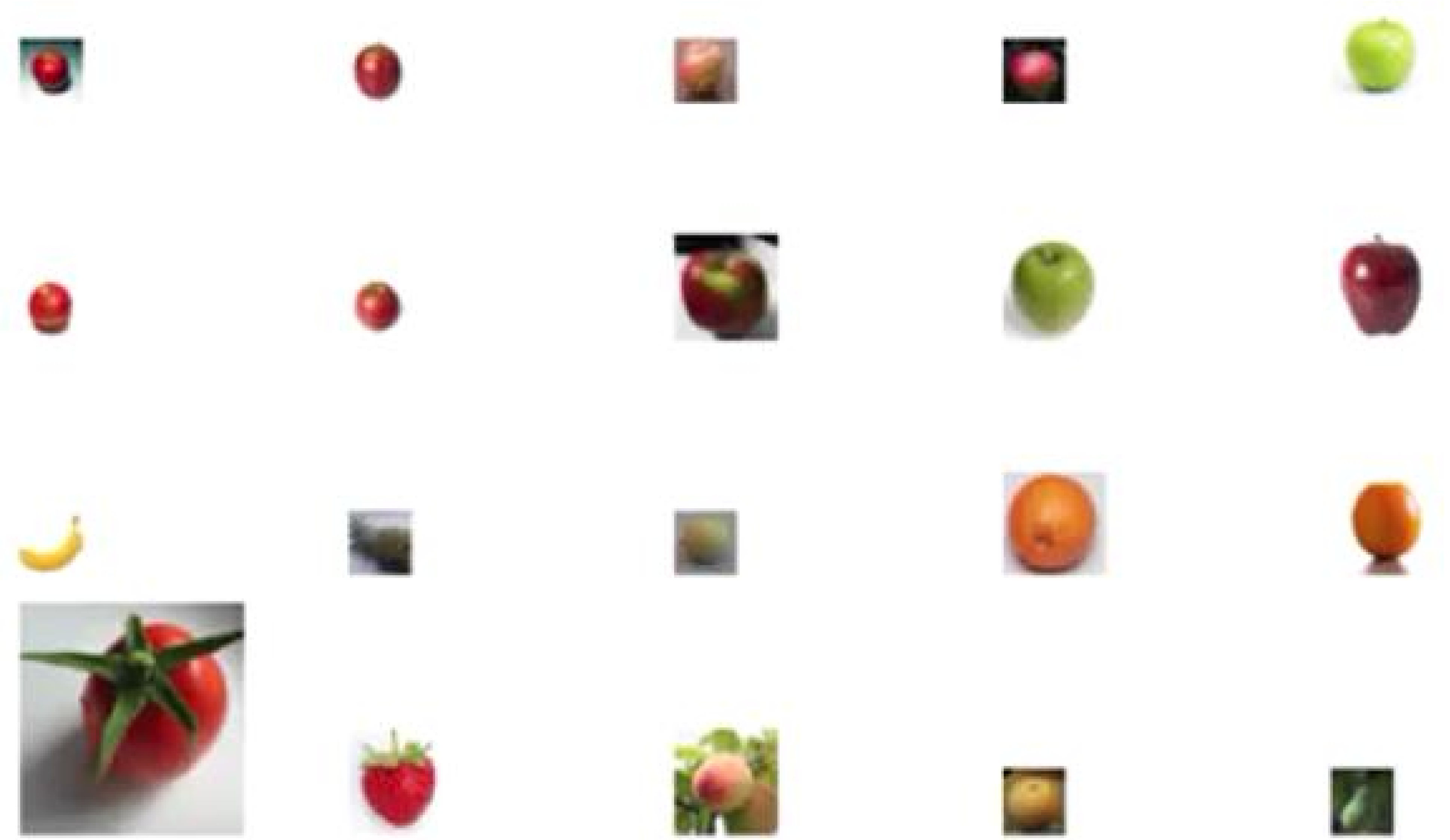
AdaBoost直观理解



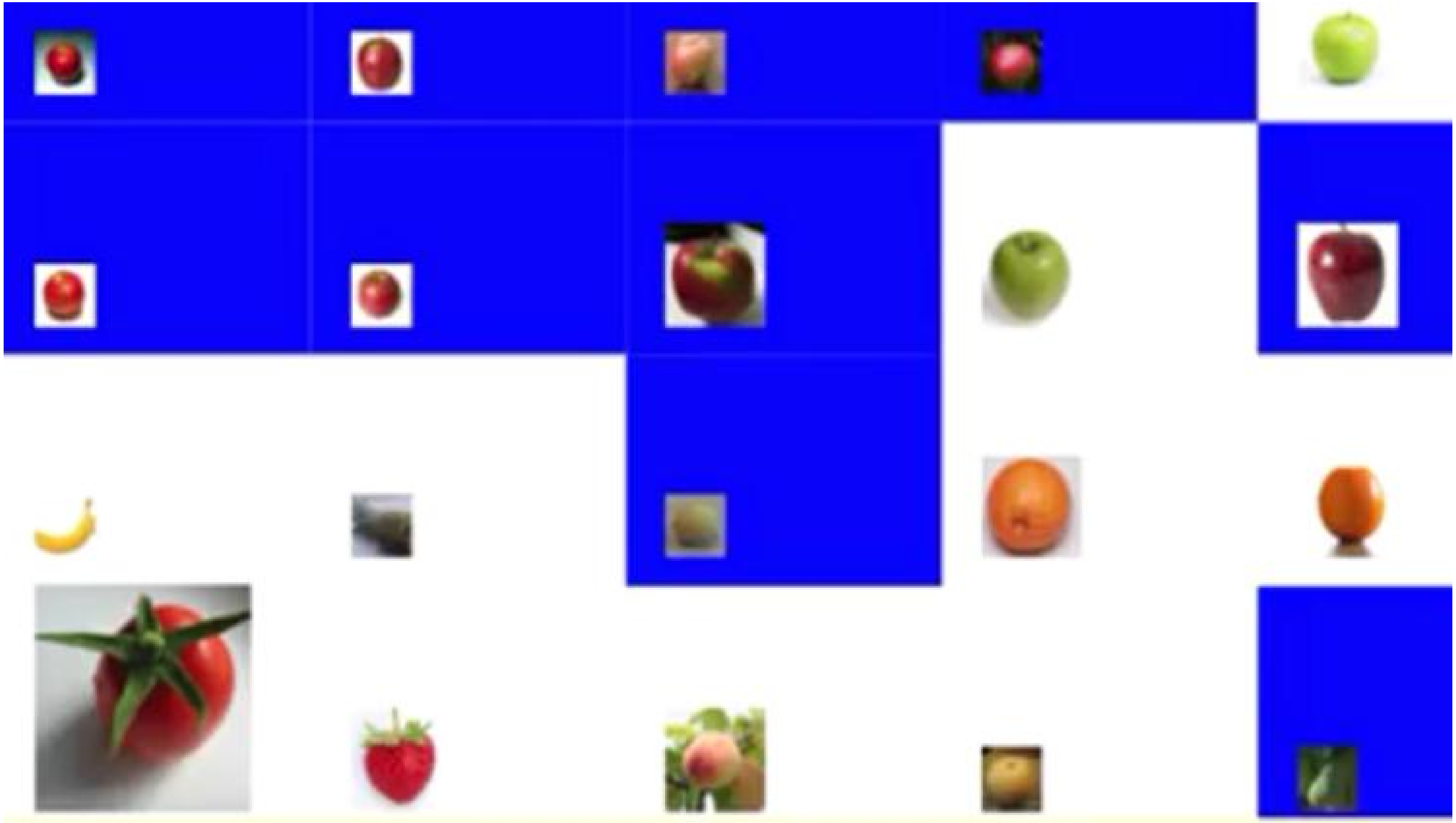
AdaBoost直观理解



AdaBoost直观理解

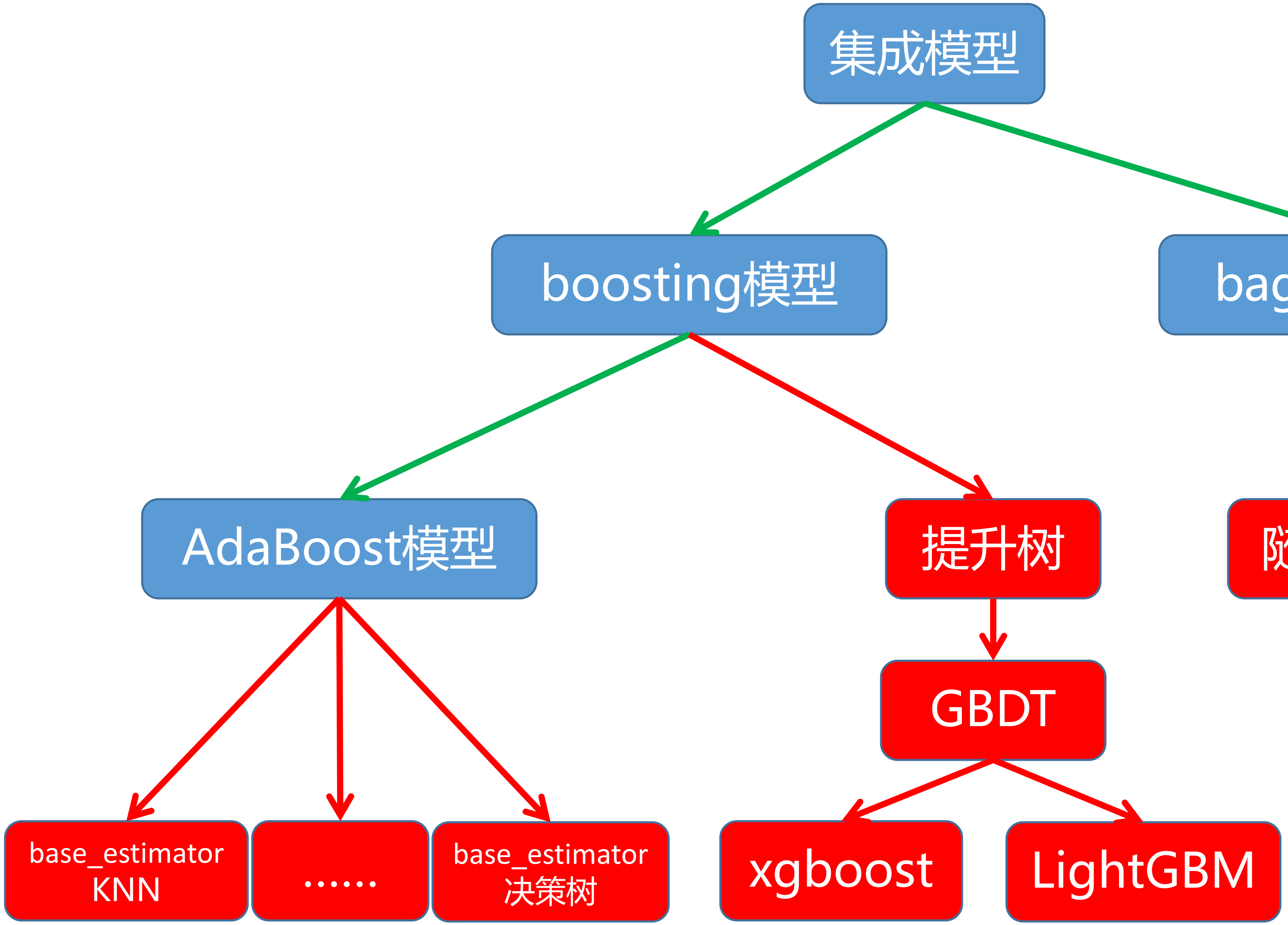


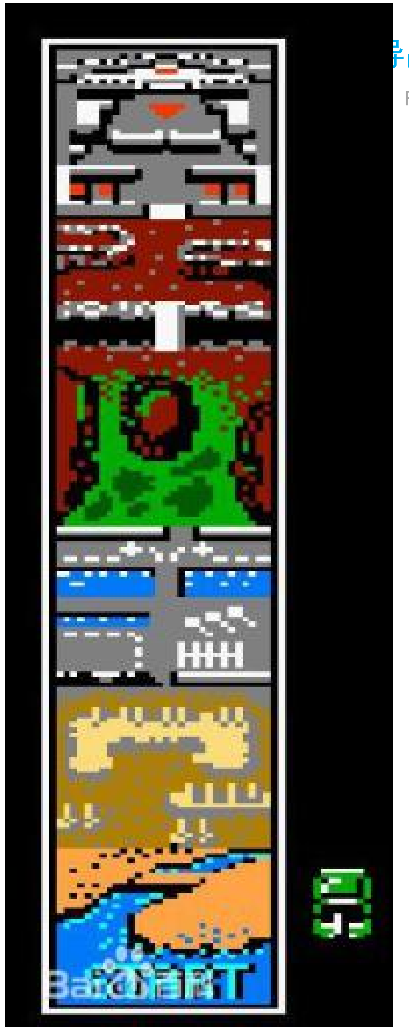
AdaBoost直观理解



AdaBoost直观理解

集成模型一览



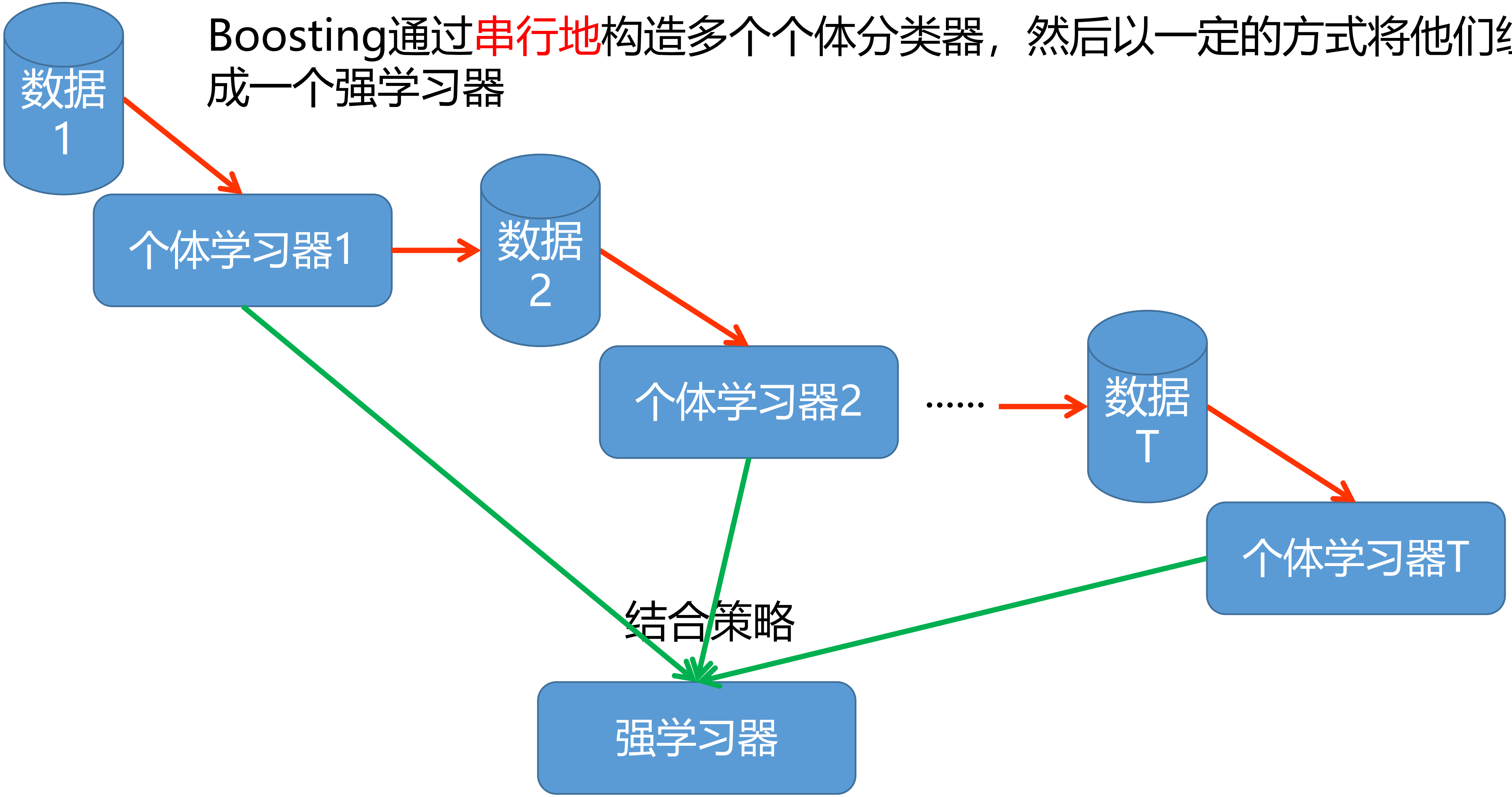


由于是抽象的，也可以叫思想

具体实现

集成学习——Boosting思想

Boosting通过串行地构造多个个体分类器，然后以一定的方式将他们组合成一个强学习器



Boosting Methods

- Boosting is one of the most powerful learning ideas introduced in the last twenty years. It was originally designed for **classification** problems, but as will be seen in this chapter, it can profitably be **extended to regression** as well. The motivation for boosting was a procedure（程序） that combines the outputs of many “weak” classifiers to produce a powerful “committee（委员会）.” From this perspective（视角） boosting bears（具有） a resemblance（相似性） to bagging and other committee-based approaches . **However** we shall see that the connection is at best（顶多） superficial（浅显的） and that boosting is fundamentally different.
- We begin by describing the most popular boosting algorithm due to Freund and Schapire called “AdaBoost.M1.”

AdaBoost（1995）思想

- 介绍
  - AdaBoost，是“Adaptive Boosting”（自适应增强）的缩写，是一种机器学习方法，由Yoav Freund和Robert Schapire于1995年提出。
- 思想
  - 前面的模型对训练集预测后，在每个样本上都会产生一个不同损失，AdaBoost会为每个样本更新权重，损失越大，样本权重越大，下一个学习器会更加“关注”这些权重大的样本；得到m个模型后，AdaBoost会根据每个模型的表现，给每个模型设置一个系数，带权叠加得到最终集成模型

AdaBoost的两个问题

- 两个问题
  - 如何得到若干个个体学习器?
  - 如何将个体学习器进行结合?

AdaBoost分类

- 输入：训练集为

$$T=\{(x_1,y_1),(x_2,y_2),\dots,(x_N,y_N)\},\qquad y\in\{-1,+1\}$$

- 训练弱学习器M个。输出：最终的强学习器G (x)
- 1) 初始化样本集权重为

$$D_1=(w_{11},w_{12},\dots,w_{1N})\quad w_{1i}=\frac{1}{N},\quad i=1,2,\dots,N$$

- 2) 对于m=1,2,...,M:
  - a) 使用具有权值分布D_m的训练数据来训练模型，得到弱学习器$G_m(x)$ ← 抽象
  - b) 计算$G_m(x)$的分类误差率

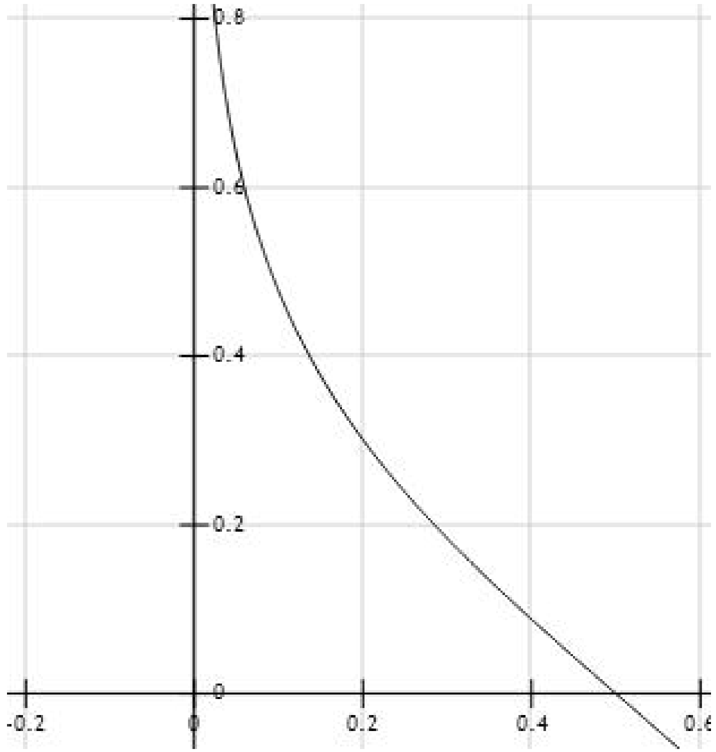
$$e_m=\sum_{i=1}^{N}w_{mi}I(G_m(x)\neq y_i)$$

AdaBoost分类

- c) 计算弱学习器的系数

$$\alpha_m = \frac{1}{2} log \frac{1 - e_m}{e_m}$$

- d) 更新训练集中样本的权重
  - 分类正确的，？？权重；分类错误的？？权重



- 3) 构建基本学习器的线性组合：

$$f(x) = \sum_{i=1}^{M} learning\_rate * \alpha_m G_m(x)$$

- 4）最终学习器为：

$$G(x) = sign(f(x))$$

AdaBoost分类*

Algorithm 10.1 *AdaBoost.M1.*

1. Initialize the observation weights $w_i = 1/N,\ i = 1, 2, \ldots, N$.
2. For $m = 1$ to M:
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute
   $$\mathrm{err}_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i}.$$
   (c) Compute $\alpha_m = \log((1 - \mathrm{err}_m)/\mathrm{err}_m)$.
   (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))],\ i = 1, 2, \ldots, N$.
3. Output $G(x) = \mathrm{sign}\left[\sum_{m=1}^M \alpha_m G_m(x)\right]$.

编程4——基于Boosting的分类

例 8.1 给定如表 8.1 所示训练数据. 假设弱分类器由 $x<v$ 或 $x>v$ 产生，其阈值 v 使该分类器在训练数据集上分类误差率最低. 试用 AdaBoost 算法学习一个强分类器.

表 8.1 训练数据表

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

AdaBoost回归*

- 输入：训练集为

$$T=\{(x_1,y_1),(x_2,y_2),\dots,(x_N,y_N)\},\qquad y\in R$$

- 训练弱学习器M个。输出：最终的强学习器G (x)
- 1) 初始化训练集权重为

$$D_1=(w_{11},w_{12},\dots,w_{1N}),\qquad w_{1i}=\frac{1}{N},i=1,2,\dots,N$$

- 2) 对于m=1,2,...,M:
  - a) 使用具有权值分布D_m的训练数据来训练模型，得到弱学习器$G_m(x)$
  - b) 计算$f_m(x)$的回归误差率

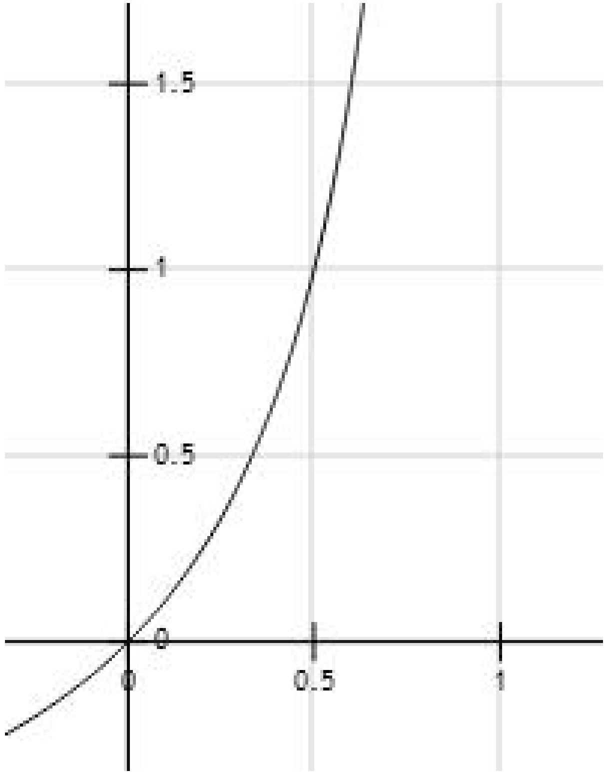
$$e_m=\sum_{i=1}^{N}w_{mi}\frac{(y_i-G_m)^2}{E_m^2}\qquad E_m=max\,|\,y_i-G_m|,i=1,2,\dots,N$$

AdaBoost回归*

- c) 计算α_m（用于计算弱学习器的系数）

$$\alpha_m = \frac{e_m}{1 - e_m}$$

- d) 更新训练集中样本的权重
  - 误差小的，降低权重，误差大的增加权重



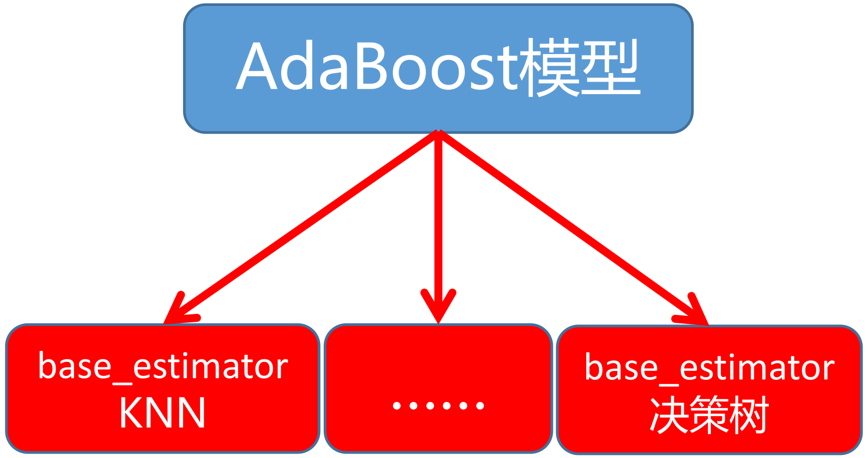
3) 最终学习器

- 先对弱学习器进行加权，取中位数，再组成最终模型

$$G(x) = \sum_{i=1}^{M} learning\_rate * (ln\frac{1}{\alpha_m})g(x)，其中g(x)是所有\alpha_m G_m(x)，m = 1,2,...,M的中位数$$

AdaBoost in sklearn

- class sklearn.ensemble.AdaBoostClassifier(base_estimator=None, n_estimators=50, learning_rate=1.0, algorithm=’ SAMME.R’ , random_state=None)
- class sklearn.ensemble.AdaBoostRegressor(base_estimator=None, n_estimators=50, learning_rate=1.0, loss=’ linear’ , random_state=None)



AdaBoost思想的优缺点

- 优点：
  - 可以使用各种回归分类模型来构建弱学习器，非常灵活
  - Sklearn中对AdaBoost的实现是从带权学习视角出发的，思想朴素，易于理解
  - 用正则、学习率、步长和控制迭代次数可以一定程度防止发生过拟合
- 缺点：
  - 对异常样本敏感，异常样本在迭代中可能会获得较高的权重，影响最终预测准确性。
  - 带权学习视角下AdaBoost做分类时无非是前向分步学习视角下的一个特例，不支持自定义损失函数

AdaBoost小结

- AdaBoost属于boosting思想，他是1995年由Freund等人提出
- 提升树
  - AdaBoost思想结合决策树的基学习器，就得到提升树模型。提升树做分类时，基学习器选CART分类树；回归时选CART回归树
- 两个视角
  - 带权学习视角、前向分布学习视角
- 前向分步学习的说明
  - 在前向分步学习视角下，当提升树的损失函数是平方损失和指数损失时，优化是简单的，但对一般损失函数而言优化难度大，即没有通用的求解方案
  - 因此2001年，Friedman提出了一个通用方案——梯度提升，起名为GBDT

编程——基于Boosting的回归

例 8.2 已知如表 8.2 所示的训练数据，x 的取值范围为区间[0.5,10.5]，y 的取值范围为区间[5.0,10.0]，学习这个回归问题的提升树模型，考虑只用树桩作为基函数.

表 8.2 训练数据表

x_i	1	2	3	4	5	6	7	8	9	10
y_i	5.56	5.70	5.91	6.40	6.80	7.05	8.90	8.70	9.00	9.05

并行地训练多颗回归树，对样本进行预测时，所有回归树同时预测，取均值作为输出

编程——基于Boosting的分类

例 8.1 给定如表 8.1 所示训练数据．假设弱分类器由 $x<v$ 或 $x>v$ 产生，其阈值 v 使该分类器在训练数据集上分类误差率最低．试用 AdaBoost 算法学习一个强分类器．

表 8.1 训练数据表

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

编程——AdaBoost综合案例之森林植被类型预测

- 数据集：
  - https://archive.ics.uci.edu/ml/datasets/covertype
- 解释：
  - 该数据集记录了美国科罗拉多州不同地块的森林植被类型。每个样本包含了描述每块土地的若干特征，包括海拔、坡度、到水源的距离、遮阳情况和土壤类型，并且随同给出了地块的已知森林植被类型。我们需要总共54 个特征中的其余各项来预测森林植被类型

Data Set Characteristics:	Multivariate	Number of Instances:	581012	Area:	Life
Attribute Characteristics:	Categorical, Integer	Number of Attributes:	54	Date Donated	1998-08-01
Associated Tasks:	Classification	Missing Values?	No	Number of Web Hits:	185453

编程——AdaBoost回归案例之共享单车租赁数量预测

- This dataset contains the hourly and daily count of rental bikes between years 2011 and 2012 in Capital bikeshare system with the corresponding weather and seasonal information.
  - 数据下载 http://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset

Data Set Characteristics:	Univariate	Number of Instances:	17389	Area:	Social
Attribute Characteristics:	Integer, Real	Number of Attributes:	16	Date Donated	2013-12-20
Associated Tasks:	Regression	Missing Values?	N/A	Number of Web Hits:	232895

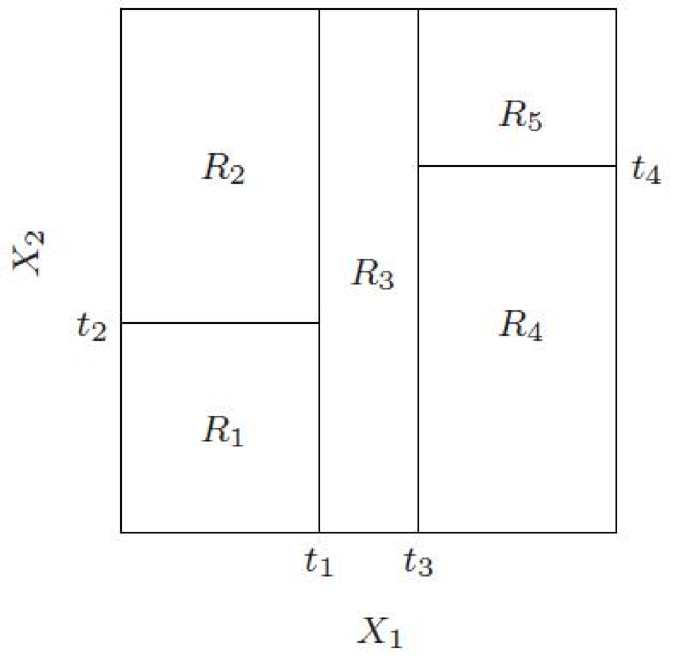
THANK YOU

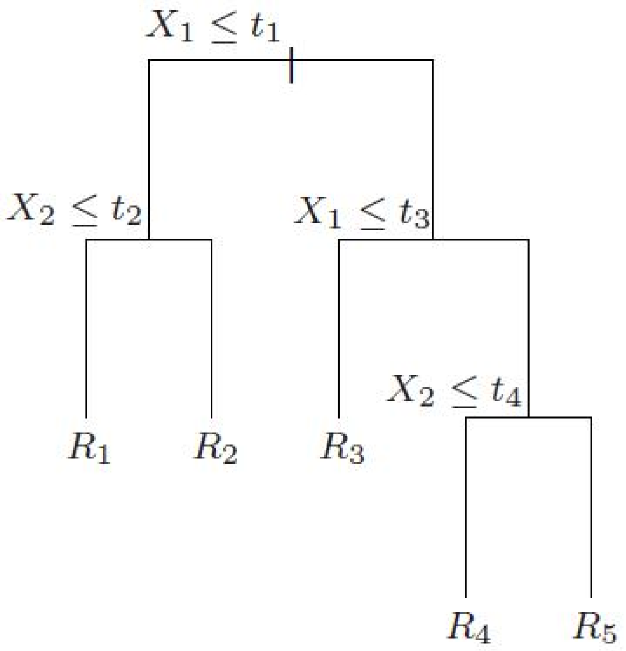
上海育创网络科技有限公司

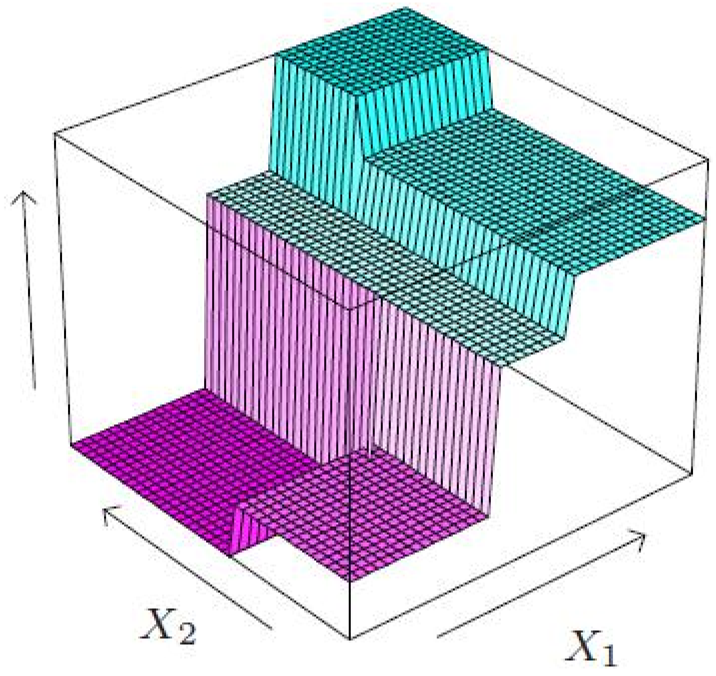
决策树参数（回归树为例）

- Suppose first that we have a partition into J regions $R_1, R_2, \ldots, R_J$, and we model the response as a constant γ_j in each region :

$$f(x) = \sum_{j=1}^{J} \gamma_j I(x \in R_j)$$







决策树参数（回归树为例）

- Regression and classification trees are discussed in detail in Section 9.2. They partition the space of all joint predictor variable values into disjoint regions R_j , j = 1, 2, . . . , J, as represented by the terminal nodes of the tree. A constant γ_j is assigned to each such region and the predictive rule is

$$x \in R_j \Rightarrow f(x) = \gamma_j$$

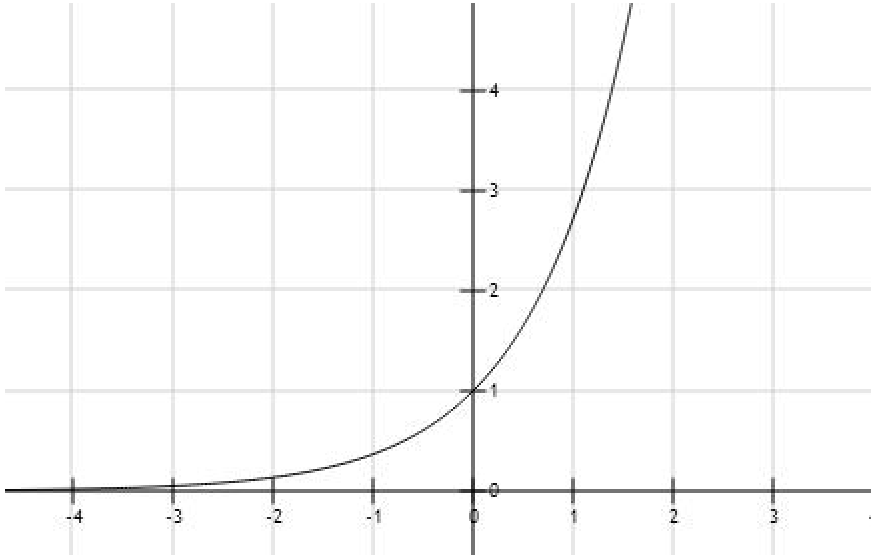
- Thus a tree can be formally expressed as

$$f(x) = \sum_{j=1}^{J} \gamma_j I(x \in R_j)$$

AdaBoost的前向分步学习视角

$$\min \sum_{i=1}^{N} L(y_i, f(x))$$

- 考虑下，$L = exp(-y_i f(x))$
- 如果y_i是+1，那$f(x)$越大越好
- 如果y_i是 -1，那$f(x)$越小越好
- 最终模型：G(x)=sign($f(x)$)



AdaBoost的前向分步学习视角*

Algorithm 10.2 *Forward Stagewise Additive Modeling.*

1. Initialize $f_0(x) = 0$.
2. For $m = 1$ to M:
   (a) Compute
   $$(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} \sum_{i=1}^{N} L(y_i, f_{m-1}(x_i) + \beta b(x_i; \gamma)).$$
   (b) Set $f_m(x) = f_{m-1}(x) + \beta_m b(x; \gamma_m)$.