实验六：集成学习方法之AdaBoost算法

PPT制作：黄行昌 詹雪莹

集成学习

- 概念：通过构建并结合多个学习器来完成学习任务。
- 一般结构：先产生一组“个体学习器”，再用某种策略将它们结合起来。个体学习器指的是使用一个现有算法，从训练数据中产生的学习器，如决策树等。
- 同质集成：只包含同种性质的学习器，如“神经网络集成”全部都是神经网络，此时的个体学习器被称为“基学习器”。
- 异质集成：包含不同类型的个体学习器，例如同时包含神经网络和决策树。此时的个体学习器被称为“组件学习器”。

集成学习

- 通过将多个学习器进行结合，常常可以获得比单一学习器显著的泛化性能，尤其是在“弱学习器”的表现上更为明显，弱学习器指的是泛化性能略优于随机猜测的学习器，例如二元分类问题上准确率略高于50%的学习器。
- 集成学习的结果一般通过多数投票产生，即少数服从多数，所以要注意的是，集成学习不一定优于单一的学习器，为了达到集成性能提高的目的，个体学习器需要“好而不同”，即个体学习器要有一点的准确性，即不能太坏，同时要具有多样性，即学习器间需要有差异。

集成学习

- 针对多样性：多样性其实就是表明学习器之间的误差是相互独立的（通俗一点理解，因为不同的原因犯错）。然而事实上，个体学习器一般是为了解决同一个问题训练出来的，它们并不能相互独立，也就是说，准确性和多样性就像“鱼和熊掌”一般，不可兼得，那么，如何产生并结合“好而不同”的学习器，就是集成学习的研究。
- 目前的几种集成学习的研究方法：1. 个体学习器之间存在强依赖关系，使用串行的方法生成，代表是Boosting；2. 不存在强依赖关系，可以使用并行方法生成，代表是Bagging和随机森林。

Boosting

- 实验只要求实现Boosting学习策略。即串行且同质。
- Boosting工作机制：先从初始训练集训练出一个基学习器，再根据基学习器的表现对训练样本分布进行调整，使先前基学习器做错的训练样本在后续受到更多关注，然后基于调整后的样本分布来训练下一个基学习器，重复上述步骤直到基学习器数目达到预设值M。最后将这M个基学习器加权结合。
- Boosting算法代表作：AdaBoost。本次实验采用最简单的实现方式，加性模型，即对基学习器进行一个线性组合。

AdaBoost

- 每一轮如何改变训练数据的权值或概率分布？
  - 提高那些被前一轮弱分类器错误分类的样本的权值，降低那些被正确分类的样本的权值
- 如何组合弱分类器？
  - AdaBoost采取加权多数表决的方法，加大误差率小的弱分类器的权值，减少分类误差率大的弱分类器的权值，并将弱分类器线性组合得到强分类器

AdaBoost

- 算法流程
  - 输入训练数据集和弱学习算法和训练轮数M，M也代表你训练的基学习器的个数
  - 初始化训练数据权重分布：

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

  - 在具有权值分布为D_m的训练数据集学习，得到基本分类器（弱分类器）：$G_m(x)$
  - 计算$G_m(x)$ 在训练数据集上的分类误差率e_m（注意计算方式是被误分类样本的权值之和）：

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

AdaBoost

- 算法流程(Cont.)
  - 计算$G_m(x)$ 的系数：

    $\alpha_m = \frac{1}{2}\log\frac{1-e_m}{e_m}$ （表示G_m分类器在最终分类器中的重要性，当$e_m \geq \frac{1}{2}$，$\alpha_m \geq 0$，并且α_m 随着e_m 减小而增大，log以e为底）
  - 更新训练数据的权值分布：

    $D_{m+1} = (w_{m+1,1},\cdots, w_{m+1,i},\cdots, w_{m+1,N})$

    $w_{m+1,i} = \frac{w_{mi}}{Z_m}\exp(-\alpha_m y_i G_m(x_i))$ （注意这里的标签y_i和$G_m(x_i)$取值为1或-1，使得误分类样本权值增大）

    Z是规范化因子：$Z_m = \sum_{i=1}^{N} w_{mi}\exp(-\alpha_m y_i G_m(x_i))$，作用是保证$D_{m+1}$是一个分布。

AdaBoost

- 算法流程(Cont.)
  - 构建基本分类器的线性组合（体现了加权表决的特性）

    $f(x) = \sum_{m=1}^{M} \alpha_m G_m(x)$
  - 输出最终分类器G(x)

    $G(x) = \text{sign}(f(x)) = \text{sign}(\sum_{m=1}^{M} \alpha_m G_m(x))$

AdaBoost算法框架

- 输入：训练集（包含N个实例），基学习算法，训练轮数（基学习器个数）
- 过程：
  - 1. 初始化训练数据权重分布$D_1 = (w_{11},\ldots, w_{1i},\ldots, w_{1N})$, $w_{1i} = \frac{1}{N}$, i = 1,2,…,N
  - 2. for m=1,2,…,M do
  - 3. 基于D_m的训练数据集学习，得到基本分类器（弱分类器）：$G_m(x)$
  - 4. 计算当前基学习器的错误率e_m，计算方法为错误样本的权重之和
  - 5. （可选：if $e_m > 0.5$ break;）
  - 6. 计算当前的基学习器的权重α_m（注意与训练数据权重区分开）
  - 7. 计算下一个基学习器的训练数据权重D_{m+1}，并且除以规范化因子Z_m
  - 8. end for
  - 9. 输出最终的分类器：$G(x) = \text{sign}(f(x)) = \text{sign}(\sum_{m=1}^{M} \alpha_m G_m(x))$

AdaBoost

- 一个简单的例子

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

  - 初始化$D_1 = \{0.1, \ldots, 0.1\}$，弱分类器由x < v 或 x > v产生
  - 阈值v取2.5的时候分类误差率最低，得到分类器：

    $G_1(x) = 1\ (x < 2.5)$

    $G_1(x) = -1\ (x > 2.5)$
  - 计算$G_1(x)$在训练数据集上的误差率：

    $e_1 = P(G_1(x_i) \neq y_i) = 0.3$

AdaBoost

- 一个简单的例子(Cont.)

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

  - 计算$G_1(x)$ 的系数：

$$\alpha_1 = \frac{1}{2}\log\frac{1-e_1}{e_1} = 0.4236$$

  - 更新训练数据权值分布：

$$w_{2,i} = \frac{w_{1i}}{Z_1}\exp(-\alpha_1 y_i G_1(x_i))$$

$D_2 = (w_{2,1},\ldots, w_{2,i},\ldots, w_{2,10}) =$
(0.07143,0.07143,0.07143,0.07143,0.07143,0.07143,
0.16667,0.16667,0.16667,0.07143)

AdaBoost

- 一个简单的例子(Cont.)

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

  - 输出分类器：

$$f_1(x) = \sum \alpha_m G_m(x) = \alpha_1 G_1(x) = 0.4236 G_1(x)$$

  - 继续迭代，重复上述步骤

AdaBoost

- 如何选择弱分类器？
  - 上述例子使用的是单层决策树（实现较简单，但是效果不一定好，可尝试两层，三层…）
  - 对于PLA，LR等强分类器，可以只随机选取一部分训练数据训练，训练出基本分类器（要注意的是选取的时候类别尽量均匀）
  - 对于其他如kNN，可以把不同的k值作为基本分类器
  - 可以尝试不同分类器的组合

实验任务与提交要求

任务：

1. 不作单独实验要求，但是要求在Project中必须体现/使用AdaBoost方法，否则倒扣百分之十的分数
2. 二分类任务中采用4种评测指标评价你的实验结果
3. 尽可能地优化与分析

剩下的实验课安排

1. 15-17周pre（验收顺序按颜色划分，非洲人先pre ）
2. 18周会进行答疑和补验收