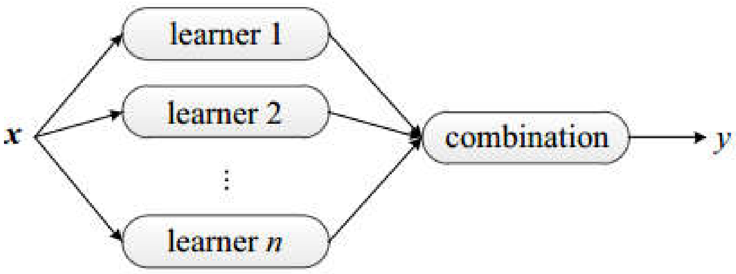
集成学习

个体与集成

通过构建并结合多个学习器来完成学习任务, 有时也被称为多分类器系统, 基于委员会的学习等.
个体学习器通常由现有的学习算法和训练数据产生.
同质集成: 集成中只包含同种类型的个体学习器, 如:决策树集成中全是决策树, 在同质的情况下, 个体学习器又称为基学习器, 个体学习算法又称为基学习算法.
异质集成: 集成中也包含不同类型的学习器, 如同时包含决策树和神经网络. 个体学习器又称为组件学习器.
集成学习示意图:



如何使得集成学习获得比任何一个个体学习器更好的性能?

当个体学习器效果相同时, 集成不起作用, 当个体学习器效果不好时, 集成起副作用, 所以集成的个体学习器应该好而不同.

假设有一个二分类问题$y \in -1, 1$, 假设基分类器的错误率为ϵ, 采用简单投票法进行集成: 设正确结果为$f(x)$预测结果为:

$$H(x) = sign(\sum_{i=1}^{T} h_i(x))$$

假设基分类器互相独立:

$$P(H(x) \neq f(x)) = \sum_{k=0}^{[T/2]} C_T^k (1-\epsilon)^k (\epsilon)^{T-k} \leq \exp(-\frac{1}{2} T (1-\epsilon)^2)$$

只要错误率不超过0.5, 随着T的增大, 总错误率会收敛到0
但是实际的个体分类器不可能完全相互独立, 这就需要我们在其多样性和准确性上有"Trade Off", 因为对于同一组数据理论上的最好模型只有一个, 所以增加多样性相当于降低了准确率, 同理增加准确率, 就要以降低多样性为代价. 如何产生好而不同的个体学习器是集成方法的研究重点.
目前的集成学习方法主要有两种:

- 个体学习器之间有较强的依赖关系, 必须采用串行胜澈过的序列化方法. 如*Boosting*
- 个体学习器之间不存在强依赖关系, 可采用同时生成的串行化方法. 如: *Bagging* 和 *Random Forest*

Boosting

Boosting是一族可以将弱学习器改变为强学习器的算法, 他的基本思想是:根据先前基学习器的表现, 调整样本数据的分布, 以使得先前的分类器分类错误的样本在后续受到更多的重视.
最著名的代表:AdaBoost算法

$$H(x)=\sum_{i=1}^{T}\alpha_t h_t(x)$$

算法描述:

输入:
训练数据集$T=\{(x_1,y_1)(x_2,y_2),\cdots,(x_N,y_N)\}, y_i\in\{-1,1\}$
*输出*最终分类器$G(x)=\{G_1,G_2,\cdots,G_n\}$的集成

1. 初始化权值分布$D_1=(w_{11},\cdots,w_{1N}), w_{1i}=\frac{1}{N}$
2. 对于$m=1,2,\cdots,m$
   a. 使用**具有权值分布D_m的训练数据集**学习, 得到基本分类器

   $$G_m(x):x\rightarrow\{-1,+1\}$$

   b. 计算$G_m(x)$在训练数据集上的分类加权误差率:

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

   c. 检查e_m是否大于0.5(是否优于随机预测)如果没有随机预测的效果好, 就结束
   d. 计算$G_m(x)$在最后集成模型中的系数(错误率越高占比越小):

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

   e. 更新训练数据集的权值分布(错误的数据提高权值, 正确的数据降低权值)

   $$w_{m+1,i}=\begin{cases}\frac{1}{Z_m}w_{m,i}e^{-\alpha_m}, G_m(x_i)=y_i\\ \frac{1}{Z_m}w_{m,i}e^{\alpha_m}, G_m(x_i)\neq y_i\end{cases}$$

3. 得到集成模型

   $$G(x)=sign[\sum_{m}^{M}\alpha_m G_m(x)]$$

   最后的结果$G(x)$的符号断定, 因为α之和未必为1

数学模型

更新公式和确定系数的公式是怎么得到的?
首先已知要将集成模型写为线性组合

$$G(x)=\sum\alpha G_m(x)$$

使用指数损失函数, 当预测值错误时loss变大, 否则loss相对较小

$$l_{exp}(\mathcal{G}|\mathcal{D}) = \mathbb{E}_{x,D}[e^{-f(x)G(x)}]$$

$$\frac{\partial l_{exp}}{\partial G} = \mathbb{E}_{x,D}[-f(x)e^{-f(x)H(x)}] = -e^{-G(x)}P(f(x)=1|x) + e^{G(x)}P(f(x)=-1|x)$$

令上式为0, 得到:

$$G(x) = \frac{1}{2}\ln\frac{P(f(x)=1|x)}{P(f(x)=-1|x)}$$

所以, 若以$sign(G(x))$作为输出, 有:

$$sign(G(x)) = sign(\frac{1}{2}\ln\frac{P(f(x)=1|x)}{P(f(x)=-1|x)}) = \begin{cases} 1, P(f(x)=1|x) \geq P(f(x)=-1|x) \\ -1, P(f(x)=1|x) \leq P(f(x)=-1|x) \end{cases}$$

所以当指数损失函数l_{exp}达到最小, 分类达到理论上的最优. 所以l_{exp}是0/1损失函数的一致性替代函数(PS: 判断是否一致替代, 知需看是否在同一个位置达到最优).

如何得到个体学习器系数α?
考虑每一个学习器$\alpha_t G_t(x)$:要找到一个α_t使得指数损失函数最小

$$\begin{aligned} l_{exp}(\alpha_t G_t|D_t) &= \mathbb{E}_{x,D}[e^{-f(x)\alpha_t G_t(x)}] \\ &= e^{-\alpha_t}P(f(x)=G_t(x)) + e^{\alpha_t}P(f(x)\neq G_t(x)) \\ &= e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t \end{aligned}$$

对α_t求导:

$$\frac{\partial l_{exp}(\alpha_t G_t|D_t)}{\partial \alpha_t} = -e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t = 0$$

解得:

$$\alpha_t = \frac{1}{2}\ln\frac{1-\epsilon_t}{\epsilon_t}$$

如何得到数据权系数D?
需要获取关于D的递推公式, 考虑我们已经得到了$t-1$个学习器的系数, 现在需要学习得到第t个学习器的系数D_t:
考虑学习器$G_t(x)$:要找到一个D_t使得指数损失函数最小, 设已经得到的$t-1$个学习器模型为$\mathcal{G}_{t-1}$, D为起始的均匀分布, 下面的损失函数为已知起始D的推论.

$$\begin{aligned} l_{exp}(\mathcal{G}_{t-1}+G_t|D) &= \mathbb{E}_{x,D}[e^{-f(x)\mathcal{G}_{t-1}(x)}e^{-f(x)G_t(x)}] \\ &= \mathbb{E}_{x,D}[e^{-f(x)\mathcal{G}_{t-1}(x)}(1-f(x)G_t(x)+\frac{f^2(x)G_t^2(x)}{2})] \\ &= \mathbb{E}_{x,D}[e^{-f(x)\mathcal{G}_{t-1}(x)}(1-f(x)G_t(x)+\frac{1}{2})] \end{aligned}$$

所以

$$\begin{aligned} G_t(x) &= arg_G \min l_{exp}(\mathcal{G}_{t-1}+G_t|D) \\ &= arg_G \min \mathbb{E}_{x,D}[e^{-f(x)\mathcal{G}_{t-1}(x)}(1-f(x)G_t(x)+\frac{1}{2})] \\ &= arg_G \max \mathbb{E}_{x,D}[e^{-f(x)\mathcal{G}_{t-1}(x)}f(x)G_t(x)] \end{aligned}$$

构造分布D_t

$$D_t(x) = \frac{D(x)e^{-f(x)\mathcal{G}_{t-1}(x)}}{\mathbb{E}_{x,D}\left(e^{-f(x)\mathcal{G}_{t-1}(x)}\right)}$$

将目标损失加一常数:

$$G_t(x) = arg_G \max \mathbb{E}_{x,D}[\frac{e^{-f(x)\mathcal{G}_{t-1}(x)}}{\mathbb{E}_{x,D}\left(e^{-f(x)\mathcal{G}_{t-1}(x)}\right)} f(x)G_t(x)] \tag{*}$$

数据权值为D_t, 每一个数据被代入到模型都是互斥的:
由全概率公式

$$\mathbb{E}_{x,D}(M(x)) = \sum_i D_i M(x_i)$$

所以(*)式:

$$G_t(x) = arg_G \max \mathbb{E}_{x,D_t}[f(x)G_t(x)]$$

易验证对于每一个x, 有:

$$f(x)h(x) = 1 - 2\mathbb{I}(f(x) \neq h(x))$$

所以这样的D_t就是要找的D_t:
下面推导递推关系:

$$\begin{aligned} D_{t+1}(x) &= \frac{D(x)e^{-f(x)\mathcal{G}_t(x)}}{\mathbb{E}_{x,D}\left(e^{-f(x)\mathcal{G}_t(x)}\right)} \\ &= \frac{D(x)e^{-f(x)\mathcal{G}_{t-1}(x)}e^{-f(x)\alpha_t \mathcal{G}_t(x)}}{\mathbb{E}_{x,D}\left(e^{-f(x)\mathcal{G}_t(x)}\right)} \\ &= D_t(x)e^{-f(x)\alpha_t \mathcal{G}_t(x)} \frac{\mathbb{E}_{x,D}\left(e^{-f(x)\mathcal{G}_{t-1}(x)}\right)}{\mathbb{E}_{x,D}\left(e^{-f(x)\mathcal{G}_t(x)}\right)} \end{aligned}$$

注意到最后一个乘积项是常数, 就是算法中的归一化系数Z_m, 与算法完全吻合.

实现方法

Boosting要求基学习器能够对特定的数据分布进行学习.
有两种方法:

- 重赋权法: 适合于能够接受权重的模型, 每轮更新权值即可
- 重采样法: 适用于不能接受权重的模型, 每轮对数据集重复采样形成新的数据集
  如果到某一轮准确率低于随即预测重赋值法直接抛弃结束, 重采样法抛弃该学习器, 并使用当前分布再次采样训练学习器.
  Boosting更注重降低偏差.

Bagging与随机森林

思想: 个体学习器应该尽量不同, 所以选择数据的不同采样训练学习器. 而数据的采样过小, 不足以进行有效学习. 所以使用交叠的采样子集.

Bagging

并行式集成学习的典型代表.
基于自助法, 获得数据子集, 然后把每一个子集训练得到的模型结合起来, 对于分类模型使用简单投票法, 对回归问题使用简单平均法

> **什么是自助法**
> 给定含m个样本的数据集D, 共选择m个数据放入采样子集中, 允许重复采样.
> 最后始终没有被采样的概率是$(1-\frac{1}{m})^m$, 取极限为$\frac{1}{e}=0.368$, 所以数据集相当大时, 一定会有40%数据漏掉, 所以成为子集.

优点:

- 不经修改可以适应多分类和回归.
- 剩余的40%左右的数据可以用于"包外分析", 用于评估泛化性能. 用剩下的数据测试准确率. 当学习器是决策树时: 可以用来辅助剪枝, 当学习器为神经网络时, 可以用来确定早停时机.

Bagging更注重降低方差., 因此适合于不剪枝决策树, 和神经网络等易受样本影响的学习器特别合适.

随机森林(RF)

bagging的扩展变体, 以决策树为基学习器构建Bagging集成, **在Bagging集成的基础上**, 在训练过程中引入随机属性. 与传统决策树不同的是, RF在每一个节点不是简单的选择最优属性, 而是在d个属性中随机选择k个, 并在k个中选取最优属性, 推荐k选择$log_2 d$.

优点:

- 降低样本扰动的同时还降低属性扰动, 进一步提升泛化性能.
- 每次只选择部分属性也有助于提升模型训练的效率.

结合策略

为什么结合?

- 统计的角度: 如果有多个假设在某一学习器上获得了相似的性能, 结合有助于降低误选率.
- 计算的角度: 帮助跳出局部最小.
- 表示的角度: 避免了单个学习器不包含全部假设空间引发的错误.

结合策略:

1. 平均法
    - 简单平均法: $H(x)=\frac{1}{T}\sum_i^T h_i(x)$
    - 加权平均法: $H(x)=\sum_i^T w_i h_i(x)$

   注意: 权重是由训练数据中习得, 由于数据存在噪声, 数据往往会过拟合, 所以加权的性能未必优于简单平均. 一般来说, 当个体学习器性能差别较大时, 采用加权法, 当个体学习器性能相近时, 采用简单平均法.
2. 投票法
   设类别集合为: $(c_1, c_2, \cdots, c_N)$
   设N个基学习器输出的向量为:$(h_i^1(x), h_i^2(x), \cdots, h_i^N(x))$表示第i个学习器在每一个类别上的概率预测.

- 绝大多数投票法:
  $H(x)=\begin{cases}c_j & \text{if}(\sum_i h_i^j(x) > 0.5\sum_k\sum_i h_i^k(x)) \\ reject & \text{otherwise}\end{cases}$
- 相对多数投票法:
  $H(x)=c_{arg_j \max \sum_i h_i^j(x)}$
- 加权投票法:
  $H(x)=c_{arg_j \max \sum_i w_i h_i^j(x)}$

说明:
绝大多数投票法提供了拒绝预测选项, 保证了可靠性. 但如果必须提供结果就必须使用相对多数投票法. 不同的学习器可能产生类型不同的预测向量:
如:
- 硬投票: 产生的预测值为01值
- 软投票: 产生的是概率值(后验概率)

不同的预测向量不能混用如何统一?
类标记(01型) + 置信度 -> 类概率, 如果学习器没有给出置信系数, 可以采用其他技术校准.

3. 学习法
   使用另一个学习器来结合学习器:
   算法描述:
   1. 生成初级学习器$H_0(x)=\{h_1,h_2,\cdots,h_N\}$
   2. 构造次级数据集$D^{'}=\Phi$
   3. 将每一个数据集$T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$中的数据代入到每一个模型中, 产生N个预测值$z_i$:组成新的数据:$((z_{i1},z_{i2},\cdots,z_{iN}),y_i)$, 加入到$D^{'}$中
   4. 根据新得到的数据集, 使用次级学习算法训练新模型得到$H(x)$.

   注意: 第三步中使用的数据集T如果和训练初级学习器使用的数据集完全相同会造成严重的过拟合. 所以每一次训练要留一部分作为T的数据.

多样性

如何使学习器具有多样性呢?

1. 误差-分歧分解
   假设使用个体学习器$\{h_1,h_2,\cdots,h_N\}$通过加权平均法得到$H(x)$, 用分歧来描述学习器的多样性
   定义学习器h_i的分歧为:

$$A(h_i|x)=(h_i(x)-H(x))^2$$

   定义集成的分歧为:

$$A(h|x)=\sum_i w_i A(h_i|x)$$

$$
\begin{aligned}
A(h|x) &= \sum_i w_i A(h_i|x) \\
&= \sum_i w_i (h_i(x) - H(x))^2 \\
&= \sum_i w_i (h_i(x)^2 - 2f(x)h_i(x) + f(x)^2 + 2f(x)h_i(x) - f(x)^2 - 2h_i(x)H(x) + H(x)^2) \\
&= \sum_i w_i (f(x) - h_i(x))^2 + \sum_i w_i (2f(x)h_i(x) - f(x)^2 - 2h_i(x)H(x) + H(x)^2) \\
&= \sum_i w_i (f(x) - h_i(x))^2 + \sum_i (2f(x)H(x) - f(x)^2 - 2H(x)^2 + H(x)^2) \\
&= \sum_i w_i (f(x) - h_i(x))^2 - \sum_i (f(x) - H(x))^2 \\
&= \sum_i w_i E(h_i|x) - E(H|x)
\end{aligned}
$$

考虑全样本, 设样本x的密度为$p(x)$,则全样本空间, 有:

$$
\sum_i^N w_i \int A(h_i|x)p(x)dx = \sum_i^N w_i \int E(h_i|x)p(x)dx - \int E(H|x)p(x)dx
$$

简记为:

$$
\sum w_i A_i = \sum w_i E_i - E
$$

集成的泛化误差就可以写作:

$$
E = \sum w_i E_i - \sum w_i A_i
$$

所以个体学习器误差越低, 学习器间的误差越小, 泛化性能越好.

2. 多样性度量
用于度量集成中个体分类器的多样性. 给定数据集$D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$, 对于二分类任务, $y_i \in \{-1, +1\}$:
设h_i, h_j预测出的样本数目如下表所示:

	$h_i = +1$	$h_j = -1$
$h_j = +1$	a	c
$h_j = -1$	b	d

$m = a + b + c + d$
多样性度量:
   - 不合度量: $\frac{b+c}{m}$
   - 相关系数: $\frac{ad-bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}$
   - Q统计量: $\frac{ad-bc}{ad+bc}$

- κ统计量$\frac{p_1-p_2}{1-p_2}$(其中:$p_1=\frac{a+d}{m}$表示两个学习器取得一致的概率, $p_2=\frac{(a+b)(a+c)}{m^2}+\frac{(c+d)(b+d)}{m^2}$表示偶然取得一致的概率(频率估计概率))

3. 多样性增强

   在训练过程中增加扰动(随机性)

   - 数据样本扰动

     如前面提到的采样法, 适合于对于输入数据敏感的学习器, 如: 决策树, 神经网络. 如果对输入数据不敏感要使用属性扰动. 如K近邻, 朴素贝叶斯, 线性回归, SVM
   - 输入属性扰动

     如随机森林算法中使用的随机子空间算法: 在数据集的属性个数很多时, 每次采用随机方法抽取部分属性, 组成属性子集训练每一个学习器, 不仅增强了多样性而且还加快了训练速度.
   - 输出表示扰动

     对训练样本类标记进行改变，如可以将分类输出转化成随机输出，可以随机改变分类样本的标记, 可以将原认务拆解为若干子任务.
   - 算法参数扰动

     基学习器的参数选择加入随机性, 比如决策树的属性选择策略, 神经网络的隐层节点个数.