发件人：Yulin Wu yw4923@nyu.edu
主题：
日期： 2020年9月9日 下午5:34
收件人：

Adaptive boosting

2020年9月6日 星期日
上午2:50

Motivation:

给出10张苹果的图片，10张不是苹果的图片，希望能教会小学生们如何辨别苹果。
老师：一开始，这20张图片都同样重要，michael你来说说看如何辨别苹果呢?
michael说：苹果是circular的。根据这个circular分类器，有些图片被分对了，有些被分错了。（如false positive, false negative)
★ 此时，全班的小学生知道了苹果是circular的。

老师：现在将被分错的图片都放大，分对的图片缩小。那么现在，nina你说说看如何辨别苹果呢?

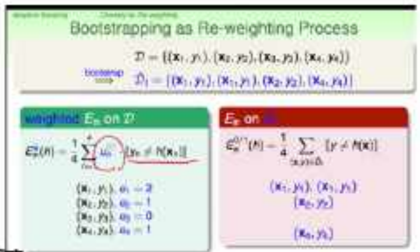
Tina：苹果是red的。根据这个red的分类器，有些图片被分对了，有些被分错了。
★ 此时，全班的小学生知道了苹果是circular的，也是red的。

老师：现在将被分错的图片都放大，分对的图片缩小。那么现在，Joey你说说看如何辨别苹果呢?

Joey：苹果可能是green的。
★ 此时，全班的小学生知道了苹果是circular的，也是red的，也可能是green的。

Boosting:

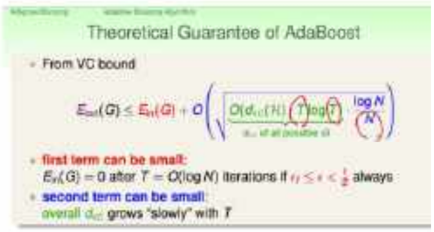
上述的过程中，一个小学生的判断就是一个分类器。老师的动作就是：每训练一个分类器，就将其犯错误的数据权重加大，来训练下一个分类器。而全班同学学到的就是这些分类器的blending，又因为这些分类器的权重相当于boostrapping，所以全班同学学到的是这些分类器的bagging。



boosting的性质:

1. 能保证分类器的diversity。
   直观来讲，g_2是在g_1做不好的数据集上做得好，那么他们肯定会挺不一样的。

2. e_in 和 e_out 相差不大。
3. 效率高，能在log(N)的时间内将e_in训练接近0.



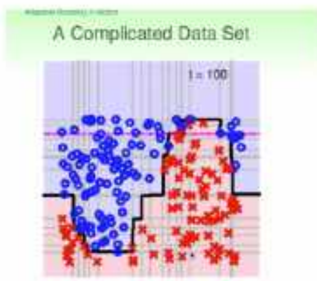
性质1的严格证明：

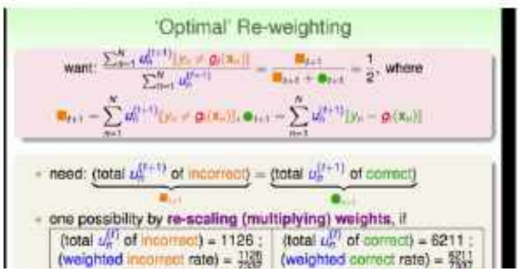
$$g_t \leftarrow \underset{h\in\mathcal{H}}{\operatorname{argmin}}\left(\sum_{n=1}^{N} u_n^{(t)} [\![y_n \neq h(\mathbf{x}_n)]\!]\right)$$

$$g_{t+1} \leftarrow \underset{h\in\mathcal{H}}{\operatorname{argmin}}\left(\sum_{n=1}^{N} u_n^{(t+1)} [\![y_n \neq h(\mathbf{x}_n)]\!]\right)$$

if g_t 'not good' for $\mathbf{u}^{(t+1)}$ $\Longrightarrow$ g_t-like hypotheses not returned as g_{t+1}

要保证g_t在g_t+1的权重数据集上表现得不好，即我要令 $\frac{\sum_{n=1}^{N} u^{t+1}[\![y_n \neq g_t(x_n)]\!]}{\sum_{n=1}^{N} u^{t+1}} = \frac{1}{2}$





则我要如何设计u^(t+1)，使得橙色方块 == 绿色圆点?
一个方法是：将犯错的数据和正确的数据的比例拉到1:1
将g_t预测错误的数据集复制 * g_t预测正确的数据集的比例（这是一个数字）倍
将g_t预测正确的数据集复制 * g_t预测错误的数据集的比例（这是一个数字）倍

incorrect: $u_n^{(t+1)} \leftarrow u_n^{(t)} \cdot 6211$ | correct: $u_n^{(t+1)} \leftarrow u_n^{(t)} \cdot 1126$

如果让ϵ_t表示犯错的比例那么

'optimal' re-weighting: let $\epsilon_t = \frac{\sum_{n=1}^N u_n^{(t)} [\![y_n \neq g_t(\mathbf{x}_n)]\!]}{\sum_{n=1}^N u_n^{(t)}}$,

multiply incorrect $\propto (1-\epsilon_t)$; multiply correct $\propto \epsilon_t$

根据这个思想，我们定义菱形_t 正比例于 1 - epsilon_t, 反比例于 epsilon_t.

Scaling Factor

'optimal' re-weighting: let $\epsilon_t = \frac{\sum_{n=1}^N u_n^{(t)} [\![y_n \neq g_t(\mathbf{x}_n)]\!]}{\sum_{n=1}^N u_n^{(t)}}$,

multiply incorrect $\propto (1-\epsilon_t)$; multiply correct $\propto \epsilon_t$

define scaling factor $\blacklozenge_t = \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$

incorrect $\leftarrow$ incorrect $\cdot \blacklozenge_t$

correct $\leftarrow$ correct $/ \blacklozenge_t$

- **equivalent** to optimal re-weighting
- $\blacklozenge_t \geq 1$ iff $\epsilon_t \leq \frac{1}{2}$
  —physical meaning: **scale up incorrect; scale down correct**
  —like what Teacher does

看到了吗！发现能够保证g_t 和g_t+1很不同的条件就是：scale up incorrect, scale down corret.

就是刚才老师做的事情，证明了boosting的性质1.

boosting algorithm 流程:

A Preliminary Algorithm

$\mathbf{u}^{(1)} = ?$

for $t = 1, 2, \ldots, T$

1. obtain g_t by $\mathcal{A}(\mathcal{D}, \mathbf{u}^{(t)})$, where $\mathcal{A}$ tries to minimize $\mathbf{u}^{(t)}$-weighted 0/1 error
2. update $\mathbf{u}^{(t)}$ to $\mathbf{u}^{(t+1)}$ by $\blacklozenge_t = \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$, where ϵ_t = weighted error (incorrect) rate of g_t

return $G(\mathbf{x}) = ?$

- want g_1 'best' for E_{in}: $u_n^{(1)} = \frac{1}{N}$
- $G(\mathbf{x})$:
  - uniform? but g_2 very bad for E_{in} (why? :-))
  - linear, non-linear? as you wish

next: a special algorithm to aggregate **linearly on the fly** with theoretical guarantee

Adaptive Boosting (AdaBoost) Algorithm

$\mathbf{u}^{(1)} = [\frac{1}{N}, \frac{1}{N}, \cdots, \frac{1}{N}]$

for $t = 1, 2, \ldots, T$

1. obtain g_t by $\mathcal{A}(\mathcal{D}, \mathbf{u}^{(t)})$, where $\mathcal{A}$ tries to minimize $\mathbf{u}^{(t)}$-weighted 0/1 error
2. update $\mathbf{u}^{(t)}$ to $\mathbf{u}^{(t+1)}$ by
   $[\![y_n \neq g_t(\mathbf{x}_n)]\!]$ (incorrect examples): $u_n^{(t+1)} \leftarrow u_n^{(t)} \cdot \blacklozenge_t$
   $[\![y_n = g_t(\mathbf{x}_n)]\!]$ (correct examples): $u_n^{(t+1)} \leftarrow u_n^{(t)} / \blacklozenge_t$
   where $\blacklozenge_t = \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$ and $\epsilon_t = \frac{\sum_{n=1}^N u_n^{(t)} [\![y_n \neq g_t(\mathbf{x}_n)]\!]}{\sum_{n=1}^N u_n^{(t)}}$
3. compute $\alpha_t = \ln(\blacklozenge_t)$

return $G(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^T \alpha_t g_t(\mathbf{x})\right)$

训练g_t

scale up incorrect, scale down correct

训练g_t+1

scale up incorrect, scale down correct

.....

最后将g blending

blending 的weight 怎么解决呢？

如果你表现好，我才将你放入我最终的blending。

菱形越大表示g这个分类器表现得越好。

- wish: large α_t for 'good' $g_t \Longleftarrow \alpha_t = \text{monotonic}(\blacklozenge_t)$
- will take $\alpha_t = \ln(\blacklozenge_t)$
  - $\epsilon_t = \frac{1}{2} \Longrightarrow \blacklozenge_t = 1 \Longrightarrow \alpha_t = 0$ (bad g_t zero weight)
  - $\epsilon_t = 0 \Longrightarrow \blacklozenge_t = \infty \Longrightarrow \alpha_t = \infty$ (super g_t superior weight)

Adaptive Boosting = weak base learning algorithm $\mathcal{A}$ (Student)
+ optimal re-weighting factor $\blacklozenge_t$ (Teacher)
+ 'magic' linear aggregation α_t (Class)