研究生算法课课堂笔记

上课日期：　　2016 年 11 月 14 日　　第(1)节课

组员学号及姓名：1601214473 马明远
组员学号及姓名：1601214476 潘翔
组员学号及姓名：1601214471 吕郁彬

内容概要：

本节课所学内容包括以下几点：

1、AdaBoost 算法的回顾

2、AdaBoost 算法的若干特点

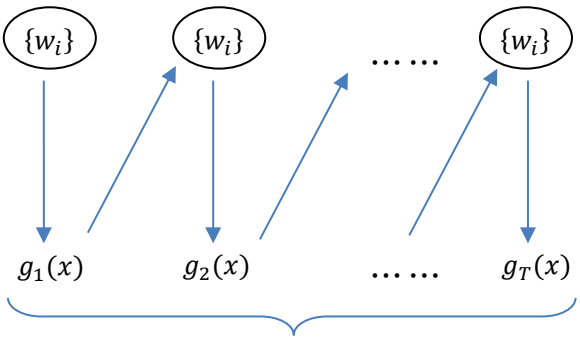
3、AdaBoost 算法的 k 分类问题

4、AdaBoost 算法与随机森林算法的比较

详细内容：

1、AdaBoost 算法的回顾

算法流程



$$G(x) = sign\left(\sum_{t=1}^{T} \ln \alpha_t \, g_t(x)\right)$$

输入：训练数据集Train $= \{(x_1, y_1)，(x_2, y_2)，\cdots，(x_N, y_N)\}$，其中$x_i \in \mathcal{X} \subseteq R^n$，$y_i \in \mathcal{Y} = \{-1, +1\}$；弱学习算法；

输出：分类器$G(x)$。

Step 1 初始化训练数据的权重

$$D = \left\{w_i | w_i = \frac{1}{N}, i = 1，2，\cdots，N\right\}$$

Step 2 对$t = 1，2，\cdots，T$，做 Step3-Step7

Step 3 使用带权重 D 的数据集进行学习，得到分类器$g_t(x)$；

Step 4 计算当前分类器的误差：

$$\varepsilon_t = \sum w_i \text{，where } g_t(x_i) \neq y_i \text{，}\left(\text{等同于 11 月 10 日课符号 } f(x^{(i)}) \neq y^{(i)}\right)$$

Step 5 计算分类器的权重更新系数

$$\alpha_t = \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$$

Step 6 更新训练数据的权重：

$$\mathrm{D} = \left\{ w_i \mid \begin{cases} w_i = w_i \cdot \alpha_t, & incorrect \\ w_i = \dfrac{w_i}{\alpha_t}, & correct \end{cases}, i = 1, 2, \cdots, N \right\}$$

Step 7 归一化处理 normalization

Step 8 计算最终分类器

$$\mathrm{G}(x) = sign\left(\sum_{t=1}^{T} \ln \alpha_t \, g_t(x)\right)$$

注：

在某些版本中，定义权重系数

$$\alpha_t = \ln \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$$

此时

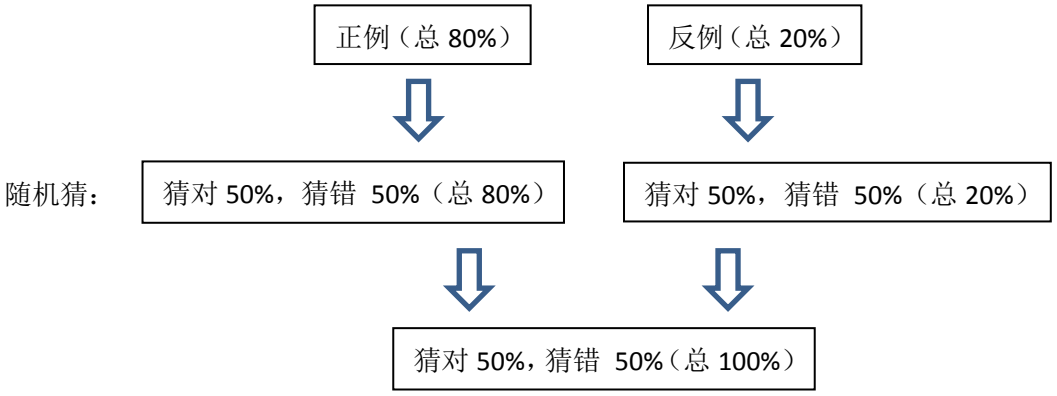
$$\mathrm{D} = \left\{ w_i \mid w_i = w_i \cdot e^{-\alpha_t y_i g_t(x_i)}, \text{其中} y_i \in \{-1, +1\}, \; g_t(x_i) \in \{-1, +1\}, \; i = 1, 2, \cdots, N \right\},$$

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

2、AdaBoost 算法的若干特点

Q2.1 在 AdaBoost 算法中，我们要求子分类器的准确率严格大于 50%。如果所给的样本分布是不均匀的，例如样本中正例占 80%，反例占 20%，问此时随意猜测能否使得准确率达到 50%?

A: 可以，子分类器的准确率与分布无关。



Q2.2 假设在学习过程中，不使用分类器，直接按照权重的大小，来猜权重大的那个，用以得到结果，是否可行？

A 不可以，假设在第一步中正例的权重占 80%，反例的权重占 20%，我们便选取正例，然后算法将猜对猜错的比例调整到各占 50%，此时正例和反例权重相同，处理过程不能继续进行下去。

Q2.3 训练数据的权重，误差等指标是调整在训练集上，还是在测试集上？

A 各项指标要在训练集上进行调整，包括权重w_i，ε_t的计算与α_t的调整都是在训练集上计算得出的。测试集用训练好的分类器计算出测试结果，不能进行各项指标调整。

Q2.4 在学习过程中每次都要把上一个分类器所得到的结果中，对的加权比例降到和错的一样大，各占 50%，为什么？

A AdaBoost 算法要实现的目的是增加子分类器的多样性。在随机森林中，通过样本以及属性的扰动来增加子分类器的差异性。而在 AdaBoost 算法中，通过调整数据的权重来实现这个目标。在每步中，将对的加权比例降到和错的一样大，各占 50%，使得下一棵树在分类之前，分错样本的总权重与分对样本的总权重均衡，通过此方法贬低上一个弱分类器的贡献，从而保证了新的分类器不受前者的影响，进而实现子分类器的多样性。此外，如果将上一轮分错的样本的权重调整的过大，下一轮过于关注上一轮被分错的样本，则不能保证弱分类器之间的干扰。

3、AdaBoost 算法的 k 分类问题

首先考虑是否可以直接将二分类问题的 AdaBoost 算法应用到多分类上。在二分类中，随机猜测一个样本，猜对和猜错的概率都是 50%，而在 k 分类时，猜对的概率将降到 1/k。此时要将子分类器的准确率都提高到 50%以上是有难度的，假设子分类器只比随机猜强一点，当前分类器的误差 ε_t将大于$1-\varepsilon_t$，这时提高权重的将是分对的那些数据。也就意味着，后续子分类将把注意力放在前面分类器分对的那些数据上，就会出现大家都学习相对简单的问题，难的那些会被放任不管。

上述问题，是因为子分类器分对的比例太小造成的，也就是说该如何调整 AdaBoost 中的α_t，使α_t>1 即可满足算法的需求，所以直接的办法就是提高分对的比例。有

$$\begin{cases} \varepsilon_t < 1-\dfrac{1}{k} \\ \dfrac{1}{k} < (1-\varepsilon_t) \end{cases} \Rightarrow (k-1)(1-\varepsilon_t) > \varepsilon_t$$

所以直观的改进方案是将 2 分类问题中的

$$\alpha_t = \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$$

更新为适合于 K 分类问题的α_t，不失一般性，K 分类问题的参数α_t对 2 分类问题依然有效。

$$\alpha_t = \sqrt{\frac{(1-\varepsilon_t)(k-1)}{\varepsilon_t}}$$

实现方法如下：

输入：k 分类训练数据集Train $= \{(x_1, y_1)，(x_2, y_2)，\cdots，(x_N, y_N)\}$，其中$x_i \in \mathcal{X} \subseteq R^n$，$y_i \in \mathcal{Y} = \{-1, +1\}$；弱学习算法；

输出：k 分类分类器$\mathrm{G}(x)$。

Step 1 初始化训练数据的权重

$$\mathrm{D} = \left\{w_i | w_i = \frac{1}{N}, i = 1，2，\cdots，N\right\}$$

Step 2 对$\mathrm{t} = 1，2，\cdots，\mathrm{T}$，做 Step3-Step7

Step 3 使用带权重 D 的数据集进行学习，得到分类器$g_t(x)$；

Step 4 计算当前分类器的误差：

$\varepsilon_t = \sum w_i$，where $g_t(x_i) \neq y_i$，（等同于 11 月 10 日课符号 $\mathrm{f(x^{(i)}) \neq y^{(i)}}$）

Step 5 计算分类器的权重更新系数

$$\alpha_t = \sqrt{\frac{(1-\varepsilon_t)(k-1)}{\varepsilon_t}}$$

Step 6 更新训练数据的权重：

$$\mathrm{D} = \left\{w_i \mid \begin{cases} w_i = w_i \cdot \alpha_t，& incorrect \\ w_i = \dfrac{w_i}{\alpha_t}，& correct \end{cases}, i = 1，2，\cdots，N\right\}$$

Step 7 归一化处理 normalization

Step 8 计算最终分类器

$$\mathrm{G}(x) = sign\left(\sum_{t=1}^{T} \ln \alpha_t \, g_t(x)\right)$$

在上述算法中，更新后对错的权重分别是：

$$\begin{cases} \sqrt{\varepsilon_t(1-\varepsilon_t)(k-1)}，& incorrect \\ \sqrt{\dfrac{\varepsilon_t(1-\varepsilon_t)}{(k-1)}}，& correct \end{cases}$$

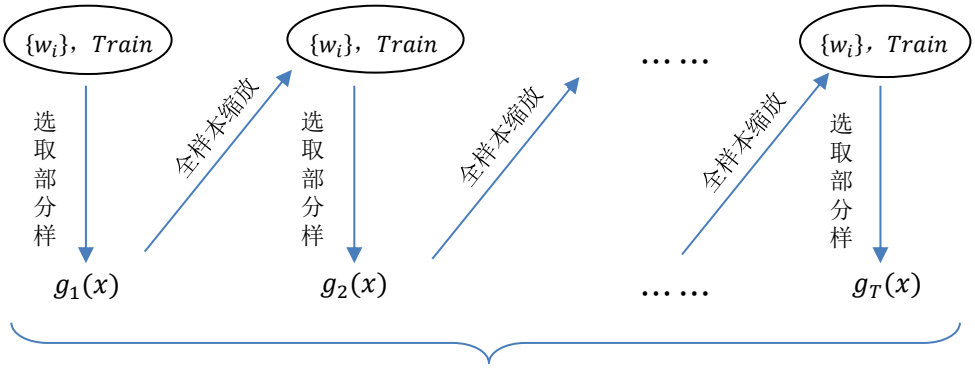
可以看到此时分对分错的比例是$1:(k-1)$，即分对的比例占总的 1/k，也就是说上一个分类器将在当前权重下达到随机猜的程度，符合 AdaBoost 的实现标准。

4、AdaBoost 算法与随机森林算法的比较

在随机森林中，我们希望得到的树是比较高的，因为该算法是大家分别学习同一个任务得到了各自的分类器，最后通过 vote-to-truth 的方式来投出最终结果。最后投票的方式不能够有效的降低 bias，所以就要求树比较高，各分类器有比较高的准确率。然而 AdaBoost 中，每一步都是对前面分错的数据进行着重处理，不断改进结果，这样做会使得 variance 难以降下来，对 bias 的降低比较有效，AdaBoost 算法通过迭代，最终必定会对数据有很高的分类正确率。AdaBoost 算法不希望子分类器太强，而是想要增加弱分类器的多样性。所以在 AdaBoost 算法则是希望树不要太高。

在随机森林中通过数据和属性的扰动来增加多样性，AdaBoost 中是否可以采用这种方法？

可以。AdaBoost 后一棵树都会对前一棵树进行补充，有效增加 ensemble 的表达能力，降低 bias。我们不希望每个子分类器有太强的分类效果，采取部分样本的采样正好实现了这个想法。具体思路是在每步中选取部分样本进行训练，然后在全体样本上对各项指标进行缩放。



$$G(x) = sign\left(\sum_{t=1}^{T} \ln \alpha_t \, g_t(x)\right)$$

综上，AdaBoost 算法通过迭代更新训练数据的权值分布，来构建一系列子分类器，最后将这些分类器进行线性组合，构成最终的强分类器。在每次迭代中，提高那些被前一轮分类器错误分类数据的权值，而降低那些被正确分类数据的权值。在最终线性组合里，分类误差率小的子分类器赋以大的权值，误差大的给以小的权值。AdaBoost 算法每次迭代可以减少当前的分类器$G(x)$在训练数据集上的分类误差率。在 AdaBoost 算法中，可以通过限制树的高度或者是采取部分采样的方法来降低子分类器的准确率，这样有效避免 over-fit。