tradaboost算法原理

定义迁移学习的模型如下：设X_b为源样例空间，X_a为辅助样例空间，源样例空间也就是我们的目标空间，就是想要去分类的样例空间。设Y={0,1}为类空间，这里简化了多分类问题为二分类问题讨论，这样我们的训练数据也就是

$$T \subseteq \{(X = X_b \cup X_a) \times Y\}$$

测试数据：

$$S = \{(x_i^t)\}, 其中 x_i^t \in X_b, 当 i = 1, 2, \ldots, k,$$

其中测试数据是未标注的，我么可以将训练数据划分为两个数据集：

$$T_a = \{(x_i^a, c(x_i^a))\}, 其中 x_i^a \in X_a, 当 i = 1, 2, \ldots, n;$$

$$T_b = \{(x_j^b, c(x_j^b))\}, 其中 x_j^b \in X_b, 当 j = 1, 2, \ldots, m.$$

其中$c(x)$代表样本数据x的真实所属的类别，T_a和T_b的区别在于T_b和测试数据S是同分布的，T_a和测试数据是不同分布的，现在的任务就是给定很少的源数据T_b和大量的辅助数据T_a训练出一个分类器在测试数据S上的分类误差最小。这里假设利用已有的数据T_b不足以训练出一个泛化能力很强的分类器。

TrAdaBoost算法

我们利用AdaBoost算法的思想原理来解决这个问题，起初给训练数据T中的每一个样例都赋予一个权重，当一个源域T_b中的样本被错误的分类之后，我们认为这个样本是很难分类的，于是乎可以加大这个样本的权重，这样在下一次的训练中这个样本所占的比重就更大了，这一点和基本的AdaBoost算法的思想是一样的。如果辅助数据集中的一个样本被错误的分类了，我们认为这个样本对于目标数据是很不同的，我们就降低这个数据在样本中所占的权重，降低这个样本在分类器中所占的比重，下面给出TradaBoost算法的具体流程：

算法 1 TrAdaBoost算法描述

输入 两个训练数据集T_a和T_b（根据公式(3.1)，合并的训练数据集$T = T_a \cup T_b$），一个未标注的测试数据集S，一个基本分类算法**Learner**，和迭代次数N。

初始化

1. 初始权重向量$\mathbf{w}^1 = (w_1^1, \ldots, w_{n+m}^1)$,[a]，其中，

$$w_i^1 = \begin{cases} 1/n & 当 i = 1, \ldots, n \\ 1/m & 当 i = n+1, \ldots, n+m \end{cases}$$

2. 设置$\beta = 1/(1 + \sqrt{2 \ln n / N})$.

For $t = 1, \ldots, N$

1. 设置$\mathbf{p}^t$满足

$$\mathbf{p}^t = \frac{\mathbf{w}^t}{\sum_{i=1}^{n+m} w_i^t}.$$

2. 调用**Learner**，根据合并后的训练数据T以及T上的权重分布$\mathbf{p}^t$和未标注数据S，得到一个在S的分类器$h_t : X \mapsto Y$。

3. 计算h_t在T_b上的错误率：

$$\epsilon_t = \sum_{i=n+1}^{n+m} \frac{w_i^t |h_t(x_i) - c(x_i)|}{\sum_{i=n+1}^{n+m} w_i^t}.$$

4. 设置$\beta_t = \epsilon_t/(1-\epsilon_t)$.[b]
5. 设置新的权重向量如下

$$w_i^{t+1} = \begin{cases} w_i^t \beta^{|h_t(x_i)-c(x_i)|}, & 当 i = 1, \ldots, n \\ w_i^t \beta_t^{-|h_t(x_i)-c(x_i)|}, & 当 i = n+1, \ldots, n+m \end{cases}$$

输出 最终分类器

$$h_f(x) = \begin{cases} 1, & \sum_{t=\lceil N/2 \rceil}^{N} \ln(1/\beta_t) h_t(x) \geq \frac{1}{2} \sum_{t=\lceil N/2 \rceil}^{N} \ln(1/\beta_t) \\ 0, & 其他 \end{cases}$$

可以看到，在每一轮的迭代中，如果一个辅助训练数据被误分类，那么这个数据可能和源训练数据是矛盾的，那么我们就可以降低这个数据的权重。具体来说，就是给数据乘上一个$\beta^{|h_t(x_i)-c(x_i)|}$,其中$\beta$的值在0到1之间，所以在下一轮的迭代中，被误分类的样本就会比上一轮少影响分类模型一些，在若干次以后，辅助数据中符合源数据的那些数据会拥有更高的权重，而那些不符合源数据的权重会降低。极端的一个情况就是，辅助数据被全部忽略，训练数据就是源数据Tb，这样这时候的算法就成了AdaBoost算法了。在计算错误率的时候，当计算得到的错误率大于0.5的话，需要将其重置为0.5。

可以看到，TrAdaBoost算法在源数据和辅助数据具有很多的相似性的时候可以取得很好效果，但是算法也有不足，当开始的时候辅助数据中的样本如果噪声比较多，迭代次数控制的不好，这样都会加大训练分类器的难度。