

# 基于自适应数据剪辑策略的 Tri-training 算法

邓 超 郭茂祖

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

**摘 要** Tri-training 能有效利用无标记样例提高泛化能力. 针对 Tri-training 迭代中无标记样例常被错误标记而形成训练集噪声, 导致性能不稳定的缺点, 文中提出 ADE-Tri-training (Tri-training with Adaptive Data Editing) 新算法. 它不仅利用 RemoveOnly 剪辑操作对每次迭代可能产生的误标记样例识别并移除, 更重要的是采用自适应策略来确定 RemoveOnly 触发与抑制的恰当时机. 文中证明, PAC 理论下自适应策略中一系列判别充分条件可同时确保新训练集规模迭代增大和新假设分类错误率迭代降低更多. UCI 数据集上实验结果表明: ADE-Tri-training 具有更好的分类泛化性能和健壮性.

**关键词** 半监督学习; 数据剪辑; 自适应策略; PAC 可学习; Tri-training

中图法分类号 TP181

## ADE-Tri-training: Tri-training with Adaptive Data Editing

DENG Chao GUO Mao-Zu

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

**Abstract** Tri-training, a Co-training style semi-supervised learning algorithm, can effectively exploit unlabeled examples to improve generalization ability. However, Tri-training may suffer more from the common problem in semi-supervised learning, i. e. the performance is usually not stable due to the unlabeled examples may often be wrongly labeled and accumulated during the iterative learning process. In this paper a new Tri-training style algorithm named ADE-Tri-training (Tri-training with Adaptive Data Editing) is proposed. ADE-Tri-training not only employs a specific Data Editing technique to identify and discard possible mislabeled examples along with iterations of three classifiers mutually labeling, but also takes an adaptive strategy to trigger or inhibit the editing operation according to different situation. The adaptive strategy is combinations of five precondition theorems all that will ensure reducing classification error as well as increasing the scale of new training set iteratively under the PAC theory. This paper also provides the proof of all these precondition theorems. Experiments on UCI datasets show that ADE-Tri-training could more effectively and stably utilize the unlabeled examples to improve classification generalization than Tri-training and DE-Tri-training (Tri-training with Data Editing but without adaptive strategy).

**Keywords** semi-supervised learning; data editing; adaptive strategy; PAC learning; Tri-training

## 1 引 言

监督学习需要大量带标记数据作训练集,以保证泛化能力.但在生物信息学和文本处理等实际应用中,对数据进行人工标记的代价很高,容易获取的是大量无标记数据.因此,将少量带标记数据和大量无标记数据结合的半监督学习成为机器学习研究热点<sup>[1]</sup>.目前,广泛研究的半监督分类算法是从监督学习角度出发,考虑带标记训练样例不足时,如何利用大量无标记样例信息辅助分类器训练<sup>[1]</sup>.已有方法包括基于 EM 算法的生成模型(generative model)参数估计法<sup>[2-3]</sup>、基于转导推理(transductive inference)的支持向量机方法<sup>[4]</sup>及 Graph-cut<sup>[5]</sup>为代表的各种图方法<sup>[6]</sup>. Blum 和 Mitchell<sup>[7]</sup>提出的 Co-training 算法是另一种半监督学习范型.该算法独立地训练两个分类器,然后采用互助方式迭代地扩充带标记样例集并重新训练.标准 Co-training 算法要求属性集能够划分为两个不相交的子集,且每个属性子集都能独立训练出分类器,这在实际问题中很难满足.为此,Goldman 等<sup>[8]</sup>提出一种改进的 Co-training 算法,不受属性集划分的约束,可用整个属性集训练两个分类器,但要求两个分类器所用的监督学习算法能够将实例空间划分为等价类集合,而且训练过程需要频繁使用耗时的交叉验证做决定. Tri-training 算法是 Zhou 等<sup>[9]</sup>提出的一种新的 Co-training 模式半监督分类算法,它使用 3 个分类器进行训练,其形式为:任意两个分类器组成联合分类器迭代地对无标记样例标记形成第 3 个分类器的新训练集,并重新训练.与标准 Co-training 算法和 Goldman 的改进算法相比, Tri-training 算法对属性集和 3 个分类器所用的监督学习算法都没有约束,而且不使用耗时的交叉验证.因此适用范围更广、效率更高.

然而,同其它半监督学习方法所遇到的问题一样,由于迭代学习过程中无标记样例常被错误地标记并积累而损害学习性能的提高<sup>[2,10]</sup>,这在 Tri-training 和其它 Co-training 模式的半监督学习中更严重<sup>[11-12]</sup>. Tri-training 训练过程是:首先由初始带标记训练集生成 3 个基分类器,然后迭代地用无标记样例对 3 个分类器精化,即如果两个分类器对一个无标记样例的标记结果一致,则该样例将被标记给另一个分类器作为新的训练样例.由于初始带标记样例数量很小,不足以训练出高精度的分类

器,所以训练过程中误标记相当数量的样例是难免的,如果在迭代过程中(尤其迭代的早期)能识别并移除这些误标记样例,则会学到更准确的假设.正如文献<sup>[9]</sup>指出:使用数据剪辑技术识别错误标记的样例是解决该问题的一种可能途径,而有效结合数据剪辑机制到 Co-training 模式半监督学习算法的研究也逐渐成为关注的焦点.例如, Li 等<sup>[13]</sup>提出的 SETRED 算法就是在 Co-training 特例算法 Self-training<sup>[14]</sup>的迭代训练过程中引入特定的数据剪辑技术来过滤自标记样例中的噪声. SETRED 所用数据剪辑技术是基于 Muhlenbach 等<sup>[15]</sup>提出的邻域图的切割边权重统计法来识别误标记样例.而据我们所知,目前针对 Tri-training 学习过程进行数据剪辑技术结合的相关研究还少有文献报道.

文献<sup>[16]</sup>中我们将基于最近邻规则的 Depuration 数据剪辑操作引入 Tri-training 过程形成 DE-Tri-training 迭代训练过程以净化每次迭代产生的新训练集中误标记噪声样例.实验表明:当初始带标记样例不足和基分类器训练不充分时,误标记的噪声样例多, Depuration 剪辑效果明显, DE-Tri-training 泛化性能较 Tri-training 有显著提高;当带标记样例充足且基分类器训练充分时,误标记的噪声样例大幅减少, Depuration 固有的剪辑错误率引起的负面效应变得明显,导致最终泛化性能反而不如 Tri-training.因此,如何在 Tri-training 训练过程中适时而非机械地执行剪辑操作,对不同情形下稳定地提高 Tri-training 算法的分类性能具有十分重要的意义.为此,本文进一步提出结合剪辑操作自适应(adaptive)策略的 Tri-training 新算法 ADE-Tri-training.该算法结合 RemoveOnly 剪辑操作和自适应数据剪辑策略到 Tri-training 学习过程,在每次迭代生成新训练集时,不仅用数据剪辑技术主动识别并移除新标记数据中可能的误标记样例以减少新训练集的噪声,同时采用的自适应策略能有效地控制不同迭代情形下 RemoveOnly 剪辑操作的触发和抑制时机,使剪辑操作固有错误率引起的负面效应不会影响分类性能迭代提高. ADE-Tri-training 中自适应策略定义为一系列判定剪辑操作触发的充分条件的组合.本文以定理形式给出不同情形下剪辑操作触发的充分条件,并证明这些充分条件在 PAC 可学习理论框架下,能够保证新训练集规模迭代增大同时确保剪辑操作的触发能进一步降低假设的分类错误率.剪辑操作的正面和负面效应分别用召回

率和剪辑规模两个因子定量刻画。

本文第 2 节简单介绍 Tri-training 基本训练过程和 RemoveOnly 剪辑操作结合方式;第 3 节证明不同情形下 RemoveOnly 剪辑操作应被触发的充分条件定理,进而定义 ADE-Tri-training 中自适应剪辑策略,这也是本文重要的理论部分;第 4 节讨论 ADE-Tri-training 算法实现;第 5 节通过实验对算法进行性能测试并对结果进行比较分析;最后对本文的研究工作进行总结。

## 2 RemoveOnly 剪辑操作的引入

文献[9]给出了 Tri-training 算法的详细伪码。作为本文新算法结合 RemoveOnly 剪辑操作的基础,这里仅描述 Tri-training 中 3 个基分类器通过互标记实现迭代训练的基本过程。

### 2.1 Tri-training 训练过程

假设初始少量带标记的样例集为  $L$ ,由  $L$  训练得到 3 个不同的初始分类器  $h_1, h_2$  和  $h_3$ ,  $x$  是无标记样例集  $U$  内任一点, Tri-training 迭代训练基本过程为:如果  $h_2$  和  $h_3$  对  $x$  的分类结果  $h_2(x)$  和  $h_3(x)$  一致,那么就可将  $x$  标记为  $h_2(x)$  并加入  $h_1$  的训练集,如此形成  $h_1$  的新训练集  $S'_1 = L \cup \{x \mid x \in U \text{ 且 } h_2(x) = h_3(x)\}$ 。类似地,  $h_2$  和  $h_3$  的训练集也分别扩充为  $S'_2$  和  $S'_3$ ,然后 3 个分类器重新训练,如此重复迭代直至  $h_1, h_2$  和  $h_3$  都没有变化,训练过程结束。

显然, Tri-training 迭代训练过程中  $h_2$  和  $h_3$  共同标记  $x$  为  $h_2(x)$ , 并给  $h_1$  作训练数据时,如果准确性足够高,会优化  $h_1$  的训练结果,否则会在  $h_1$  的训练集中加入噪声,影响训练效果。为此, Zhou 等<sup>[9]</sup> 分析得出一个能使假设分类错误率迭代降低的充分条件,并以该充分条件作为判断准则来决定新标记的样例集是否应该被加入新训练集。作为自适应剪辑策略中的一种基本情形,本文将在第 3 节对该充分条件以定理形式(定理 1)重新描述并证明。

### 2.2 RemoveOnly 剪辑操作的结合

文献[16]的 DE-Tri-training 算法中,我们采用基于最近邻规则的 Depuration 技术对新训练集剪辑, Depuration 是第一种应用原型选择策略的技术,可分解为 RemoveOnly 和 RelabelOnly 两个子剪辑操作, RemoveOnly 从训练集移除“可疑”样例而 RelabelOnly 仅对样例的错误标记修正<sup>[17]</sup>。因为 Jiang 等<sup>[17]</sup> 通过实验表明: Depuration 的剪辑效果仅与

RelabelOnly 相当,且二者都没有 RemoveOnly 效果好,所以 ADE-Tri-training 新算法只选用 RemoveOnly 子操作来识别并移除新标记训练集中可疑噪声,具体过程为:对新标记训练集  $L'$  中每个样例,首先按最近邻规则从  $L \cup L'$  中选取它的  $k$  个近邻,然后观察其中是否有  $k'$  个近邻的标记相同,若没有,则该新样例被识别为“可疑”的误标记样例,将从  $L'$  中移除。文献[18]指出当  $k$  和  $k'$  设为 3 和 2 时,实际剪辑效果最好, ADE-Tri-training 采用此设定。

ADE-Tri-training 仅选用 Depuration 的 RemoveOnly 子操作对可疑样例移除,而拒绝 RelabelOnly 子操作根据  $k'$  个近邻的相同标记对新样例标记进行修改。除能得到好的剪辑效果外,还有另外两方面优点:(1)那些可能被误标记的样例被识别后,简单地抛弃这些样例,而不会尝试重新标记,这样可避免引入新的噪声到训练集。(2)相对简单的 RemoveOnly 子操作使剪辑性能的定量刻画更容易,有利于自适应策略的制定与实现。

与 DE-Tri-training 对新训练集机械执行剪辑操作的做法不同, ADE-Tri-training 中定义了自适应策略灵活控制不同迭代情形下 RemoveOnly 剪辑操作的恰当触发时机,称为自适应剪辑策略。

## 3 自适应剪辑策略

自适应策略是基于 Angluin 和 Laird<sup>[19]</sup> 有关训练集带噪声时目标假设的 PAC 可学习性结论构造的。本节首先描述 Tri-training 使分类错误率迭代降低的训练集更新充分条件,然后对剪辑操作性定量刻画,最后证明引入剪辑操作后假设分类错误率迭代降低的一系列充分条件,定义自适应剪辑策略。

### 3.1 考虑训练集噪声的 PAC 可学习性

按照 Angluin 和 Laird<sup>[19]</sup> 的结论:当抽取含  $m$  个训练样例的序列  $\sigma$  时,如果样例规模  $m$  满足条件  $m \geq \frac{2}{\xi^2(1-2\eta)^2} \ln\left(\frac{2N}{\delta}\right)$ , 则与序列  $\sigma$  不一致程度最小的假设  $H_i$  具有 PAC 可学习性,即  $Pr[d(H_i, H^*) \geq \xi] \leq \delta$ 。其中,  $\xi$  是最坏情况下假设的分类错误率,  $\eta (< 0.5)$  是训练集上噪声率的上限,  $N$  是假设的数目,  $\delta$  是置信度,  $d(H_i, H^*)$  是假设  $H_i$  和真实假设  $H^*$  相应样例集的对称差集中元素概率和,即  $Pr(x \in H_i \cup H^* \text{ 且 } x \notin H_i \cap H^*)$ 。

令  $m' = \frac{2}{\xi^2(1-2\eta)^2} \ln\left(\frac{2N}{\delta}\right)$  且  $\mu = m/m'$ , 则  $m =$

$\frac{2\mu}{\xi^2(1-2\eta)^2} \ln\left(\frac{2N}{\delta}\right)$ , 进而令常量  $c = 2\mu \ln\left(\frac{2N}{\delta}\right)$ , 则有  $m = \frac{c}{\xi^2(1-2\eta)^2}$ . 所以, 学习器训练所得假设的分类错误率与训练集规模、训练集噪声率间有如下关系成立(记为  $u$ ):

$$u = \frac{c}{\xi^2} = m(1-2\eta)^2 \quad (1)$$

显然, 式(1)蕴含着  $u \propto \frac{1}{\xi^2}$ .

Tri-training 和 ADE-Tri-training 都是基于式(1), 保证每个分类器新训练集规模  $m$  迭代增大的同时, 确保每次重新训练所得假设的分类错误率  $\xi$  迭代地降低. 下面仍以  $h_2$  和  $h_3$  标记新样例给  $h_1$  提供新训练集的迭代过程为代表进行分析.

### 3.2 标准 Tri-training 训练集更新充分条件

Tri-training 每一轮迭代中,  $h_2$  和  $h_3$  从  $U$  中选择某些样例标记给  $h_1$ . 由于 Tri-training 过程中 3 个分类器都被重新训练, 所以不同轮次中被标记并选中加入新训练集的无标记样例的数目和具体组成不同. 令  $L^t$  和  $L^{t-1}$  分别代表第  $t$  轮和第  $t-1$  轮为  $h_1$  新标记的样例集, 且第  $t$  轮迭代时,  $L^{t-1}$  中所有样例会被当作无标记样例重新放回  $U$  中, 则第  $t$  轮和第  $t-1$  轮新训练集分别为  $L \cup L^t$  和  $L \cup L^{t-1}$ , 规模大小  $m^t$  和  $m^{t-1}$  分别为  $|L \cup L^t|$  和  $|L \cup L^{t-1}|$ . 令  $\eta_L$  代表  $L$  上的噪声率, 则  $L$  中误标记样例数目为  $\eta_L |L|$ . 令  $\bar{e}_1^t (< 0.5)$  代表第  $t$  轮时  $h_2$  和  $h_3$  组成的联合分类器  $h_2$  和  $h_3$  的分类错误率上限, 即  $h_2$  和  $h_3$  给出一致标记的样例中错误标记所占比例, 则  $L^t$  中误标记样例数目为  $\bar{e}_1^t |L^t|$ . 由此可得, 第  $t$  轮迭代  $h_1$  的新训练集噪声率为

$$\eta^t = \frac{\eta_L |L| + \bar{e}_1^t |L^t|}{|L \cup L^t|} \quad (2)$$

**定理 1.** PAC 可学习框架下, Tri-training 相邻两轮迭代( $t-1$  轮和  $t$  轮,  $t > 1$ )中, 当  $0 < \frac{\bar{e}_1^t}{\bar{e}_1^{t-1}} < \frac{|L^{t-1}|}{|L^t|} < 1$  成立时, 用  $L \cup L^{t-1}$  和  $L \cup L^t$  对  $h_1$  先后训练所得假设的分类错误率  $\xi^{t-1}$  和  $\xi^t$  具有性质  $\xi^t < \xi^{t-1}$ , 即  $h_1$  的分类准确率会迭代提高.

证明. PAC 可学习框架下, Tri-training 两轮迭代对应式(1)分别为  $u^t = \frac{c}{(\xi^t)^2} = m^t(1-2\eta^t)^2$  和  $u^{t-1} = \frac{c}{(\xi^{t-1})^2} = m^{t-1}(1-2\eta^{t-1})^2$ . 两轮迭代

新训练集噪声率  $\eta^t$  和  $\eta^{t-1}$  对应式(2)分别为  $\eta^t = \frac{\eta_L |L| + \bar{e}_1^t |L^t|}{|L \cup L^t|}$  和  $\eta^{t-1} = \frac{\eta_L |L| + \bar{e}_1^{t-1} |L^{t-1}|}{|L \cup L^{t-1}|}$ .

$$0 < \frac{\bar{e}_1^t}{\bar{e}_1^{t-1}} < \frac{|L^{t-1}|}{|L^t|} < 1 \text{ 成立等价于 } |L^t| > |L^{t-1}| > 0$$

和  $\bar{e}_1^t |L^t| < \bar{e}_1^{t-1} |L^{t-1}|$  同时成立. 由  $|L^t| > |L^{t-1}|$  可推得  $|L \cup L^t| > |L \cup L^{t-1}|$ , 即  $m^t > m^{t-1}$ , 同时结合  $\bar{e}_1^t |L^t| < \bar{e}_1^{t-1} |L^{t-1}|$  可推得  $\eta^t < \eta^{t-1}$ , 进而可得  $m^t(1-2\eta^t)^2 > m^{t-1}(1-2\eta^{t-1})^2$ , 即  $u^t > u^{t-1}$ , 由于  $u \propto \frac{1}{\xi^2}$ , 所以  $\xi^t < \xi^{t-1}$ . 证毕.

文献[9]中, Tri-training 迭代训练过程正是以定理 1 充分条件  $0 < \frac{\bar{e}_1^t}{\bar{e}_1^{t-1}} < \frac{|L^{t-1}|}{|L^t|} < 1$  作为训练集更新的判断准则. 先判定  $0 < \frac{\bar{e}_1^t}{\bar{e}_1^{t-1}} < 1$  和  $0 < \frac{|L^{t-1}|}{|L^t|} < 1$  成立, 再判定  $\frac{\bar{e}_1^t}{\bar{e}_1^{t-1}} < \frac{|L^{t-1}|}{|L^t|}$  即  $\bar{e}_1^t |L^t| < \bar{e}_1^{t-1} |L^{t-1}|$  同时成立即可.

ADE-Tri-training 引入 RemoveOnly 剪辑操作后, 还需要关注剪辑操作对定理 1 中充分条件的影响. 该充分条件要求每轮迭代产生的新标记训练集同时满足误标记率  $\bar{e}_1^t$  和规模  $|L^t|$  两方面约束, 才能保证重新训练所得假设的泛化能力迭代提高. RemoveOnly 操作把误标记引起的噪声率  $\bar{e}_1^t$  降低而发挥正面效应的同时, 对可疑噪声的移除会使  $|L^t|$  减小形成负面效应. 如果 RemoveOnly 固有的剪辑误差率使负面效应明显大于正面效应, 则极可能导致定理 1 充分条件不再满足. 因此, 采用合理的指标精确度量剪辑操作的正面和负面效应, 进而制定自适应策略控制不同情形下 RemoveOnly 的触发和抑制是十分必要的.

### 3.3 RemoveOnly 性能定量刻画指标

ADE-Tri-training 中 RemoveOnly 操作对新标记训练集  $L^t$  的剪辑性能用误标记样例的召回率  $r^t$  和剪辑后剩余新训练集  $L_{de}^t$  的规模  $|L_{de}^t|$  两个因子来度量.

召回率  $r^t$  定义如下:

$$r^t = \frac{\text{RemoveOnly 正确移除的误标记样例数}}{L^t \text{ 中实际误标记样例数}} \quad (3)$$

直观上, 召回率  $r^t$  反映了 RemoveOnly 对  $L^t$  中误标记引起的噪声率  $\bar{e}_1^t$  的降低程度, 而  $|L_{de}^t|$  则反映了  $|L^t|$  减小的程度. 这样 RemoveOnly 剪辑操作对定理 1 中充分条件的正面和负面效应就可以用这两

个因子精确刻画。

### 3.4 RemoveOnly 触发的充分条件

与式(2) $L \cup L'$ 的噪声率  $\eta'$  相对应, RemoveOnly 对  $L'$  剪辑后  $L \cup L'_{de}$  的噪声率  $\eta'_{de}$  为

$$\eta'_{de} = \frac{\eta_L |L| + (1-r')\bar{e}'_1 |L'|}{|L \cup L'_{de}|} \quad (4)$$

与第  $t$  轮迭代已得到的  $m'$  和  $\xi'$  相对应, 令  $m'_{de}$  为  $L \cup L'_{de}$  的规模, 即  $|L \cup L'_{de}|$ . 令  $\xi'_{de}$  为  $L \cup L'_{de}$  对  $h_1$  训练所得假设的分类错误率。

下面的定理 2 刻画了同一轮迭代中触发 RemoveOnly 使分类准确率提高的充分条件, 定理 3~定理 5 刻画了上一轮迭代 RemoveOnly 触发或抑制的不同情形下分类准确率迭代提高的充分条件, 这一系列充分条件将构成自适应策略中 RemoveOnly 触发与否的基本判定条件。

**定理 2.** PAC 可学习框架下, ADE-Tri-training 第  $t(t \geq 1)$  轮迭代中, RemoveOnly 对  $L'$  剪辑得到  $L'_{de}$ , 则当  $|L'_{de}| < |L'|$  且  $r' \geq \frac{|L'| - |L'_{de}|}{2\bar{e}'_1 |L'|}$  成立时, 用  $L \cup L'^{-1}$  和  $L \cup L'_{de}$  对  $h_1$  分别训练所得假设的分类错误率  $\xi'$  和  $\xi'_{de}$  具有性质  $\xi'_{de} < \xi'$ 。

证明. 由  $m' = |L \cup L'| = |L| + |L'|$  和  $m'_{de} = |L \cup L'_{de}| = |L| + |L'_{de}|$  代入式(2)和式(4)得剪辑前后噪声率分别为  $\eta' = \frac{\eta_L |L| + \bar{e}'_1 |L'|}{m'}$  和  $\eta'_{de} = \frac{\eta_L |L| + (1-r')\bar{e}'_1 |L'|}{m'_{de}}$ , 由此 PAC 可学习框架下的  $u'$  和  $u'_{de}$  分别为  $u' = m'(1 - 2\eta')^2 = \frac{(m' - 2\eta_L |L| - 2\bar{e}'_1 |L'|)^2}{m'}$  和  $u'_{de} = m'_{de}(1 - 2\eta'_{de})^2 = \frac{(m'_{de} - 2\eta_L |L| - 2(1-r')\bar{e}'_1 |L'|)^2}{m'_{de}}$ , 由  $|L'_{de}| < |L'|$  得  $m'_{de} < m'$ . 结合已有假设  $\eta_L < 0.5$ ,  $\bar{e}'_1 < 0.5$  容易推得  $r' \geq \frac{|L'| - |L'_{de}|}{2\bar{e}'_1 |L'|}$  等价于  $m'_{de} - 2\eta_L |L| - 2(1-r')\bar{e}'_1 |L'| \geq m' - 2\eta_L |L| - 2\bar{e}'_1 |L'| > 0$ , 进而可得  $u'_{de} > u'$ , 由于  $u \propto \frac{1}{\xi^2}$ , 所以  $\xi'_{de} < \xi'$ . 证毕。

**定理 3.** PAC 可学习框架下, ADE-Tri-training 相邻两轮迭代( $t-1$  轮和  $t$  轮,  $t > 1$ )中, 第  $t-1$  轮 RemoveOnly 未触发, 且  $0 < \frac{|L'^{-1}|}{|L'|} < 1$  情形下, 当  $|L'^{-1}| < |L'_{de}| < |L'|$  且  $0 < \frac{(1-r')\bar{e}'_1}{\bar{e}'_1 |L'|} \leq \frac{|L'^{-1}|}{|L'|} < 1$  成立时, 用  $L \cup L'^{-1}$  和  $L \cup L'_{de}$  对  $h_1$  分别训练所得假

设的分类错误率  $\xi'^{-1}$  和  $\xi'_{de}$  具有性质  $\xi'_{de} < \xi'^{-1}$ 。

证明. PAC 可学习框架下, 分别有  $u'^{-1} = \frac{c}{(\xi'^{-1})^2} = m'^{-1}(1 - 2\eta'^{-1})^2$  和  $u'_{de} = \frac{c}{(\xi'_{de})^2} = m'_{de}(1 - 2\eta'_{de})^2$ . 噪声率分别为  $\eta'^{-1} = \frac{\eta_L |L| + \bar{e}'_1 |L'^{-1}|}{m'^{-1}}$  和  $\eta'_{de} = \frac{\eta_L |L| + (1-r')\bar{e}'_1 |L'|}{m'_{de}}$ . 由  $|L'^{-1}| < |L'_{de}|$  得  $m'_{de} > m'^{-1}$ , 同时容易验证  $r' \geq 1 - \frac{\bar{e}'_1 |L'^{-1}|}{\bar{e}'_1 |L'|} \geq 0$  与  $(1-r')\bar{e}'_1 |L'| \leq \bar{e}'_1 |L'^{-1}|$  等价, 由此得  $\eta'_{de} \leq \eta'^{-1}$ , 进而推得  $u'_{de} > u'^{-1}$ , 由于  $u \propto \frac{1}{\xi^2}$ , 所以  $\xi'_{de} < \xi'^{-1}$ . 证毕。

**定理 4.** PAC 可学习框架下, ADE-Tri-training 相邻两轮迭代( $t-1$  轮和  $t$  轮,  $t > 1$ )中, 第  $t-1$  轮 RemoveOnly 被触发情形下, 当  $|L'| > |L'_{de}| > 0$  且  $0 < \frac{\bar{e}'_1}{(1-r'^{-1})\bar{e}'_1 |L'|} \leq \frac{|L'^{-1}|}{|L'|} < 1$  成立时, 用  $L \cup L'_{de}$  和  $L \cup L'$  对  $h_1$  先后训练所得假设的分类错误率  $\xi'^{-1}$  和  $\xi'$  具有性质  $\xi' < \xi'^{-1}$ 。

证明. PAC 可学习框架下, 分别有  $u'^{-1} = \frac{c}{(\xi'^{-1})^2} = m'^{-1}(1 - 2\eta'^{-1})^2$  和  $u' = \frac{c}{(\xi')^2} = m'(1 - 2\eta')^2$ , 噪声率分别为  $\eta'^{-1} = \frac{\eta_L |L| + (1-r'^{-1})\bar{e}'_1 |L'^{-1}|}{m'^{-1}}$  和  $\eta' = \frac{\eta_L |L| + \bar{e}'_1 |L'|}{m'}$ . 由  $|L'| > |L'_{de}|$  得  $m' > m'^{-1}$ , 结合  $0 < \frac{\bar{e}'_1}{(1-r'^{-1})\bar{e}'_1 |L'|} < \frac{|L'^{-1}|}{|L'|} < 1$  成立等价于  $|L'| > |L'^{-1}| > 0$  和  $\bar{e}'_1 |L'| < (1-r'^{-1})\bar{e}'_1 |L'^{-1}|$  同时成立, 可推得  $\eta' < \eta'^{-1}$ , 进而可得  $u' > u'^{-1}$ , 由于  $u \propto \frac{1}{\xi^2}$ , 所以  $\xi' < \xi'^{-1}$ . 证毕。

**定理 5.** PAC 可学习框架下, ADE-Tri-training 相邻两轮迭代( $t-1$  轮和  $t$  轮,  $t > 1$ )中, 第  $t-1$  轮 RemoveOnly 被触发且  $0 < \frac{|L'_{de}|}{|L'|} < 1$  情形下, 当  $|L'_{de}| > |L'^{-1}| > 0$  且  $0 < \frac{(1-r')\bar{e}'_1}{(1-r'^{-1})\bar{e}'_1 |L'|} \leq \frac{|L'^{-1}|}{|L'|} < 1$  成立时, 用  $L \cup L'_{de}$  和  $L \cup L'^{-1}$  对  $h_1$  先后训练所得假设的分类错误率  $\xi'^{-1}$  和  $\xi'_{de}$  具有性质  $\xi'_{de} < \xi'^{-1}$ 。

证明. 由  $|L'_{de}| > |L'^{-1}|$  得  $m'_{de} > m'^{-1}$ , 同时  $0 < \frac{(1-r')\bar{e}'_1}{(1-r'^{-1})\bar{e}'_1 |L'|} \leq \frac{|L'^{-1}|}{|L'|} < 1$  等价于  $(1-r')\bar{e}'_1 |L'| \leq$

$(1-r^{t-1})\bar{e}_1^{t-1}|L^{t-1}|$ , 代入式(4)和式(1)推得  $\eta_{de}^t < \eta_{de}^{t-1}$  和  $u_{de}^t > u_{de}^{t-1}$ , 由  $u \propto \frac{1}{\xi^2}$  得  $\xi_{de}^t < \xi_{de}^{t-1}$ . 证毕.

### 3.5 自适应剪辑策略

ADE-Tri-training 的自适应剪辑策略正是利用定理 1~定理 5 的充分条件来断定第  $t$  轮迭代中哪些情形下 RemoveOnly 应该被触发或抑制, 具体定义为在第  $t$  轮迭代时,

(1) 若第  $t-1$  轮迭代中 RemoveOnly 未被触发且  $|L^t| > |L^{t-1}|$ , 则定理 1 使  $\xi^t < \xi^{t-1}$  和定理 2 使  $\xi_{de}^t < \xi^t$  的充分条件都能满足的情形下(保证  $\xi_{de}^t < \xi^t < \xi^{t-1}$ ), RemoveOnly 将被触发.

(2) 若第  $t-1$  轮迭代中 RemoveOnly 未被触发且  $|L^t| > |L^{t-1}|$ , 则在定理 1 使  $\xi^t < \xi^{t-1}$  的充分条件无法满足, 但定理 3 使  $\xi_{de}^t < \xi^{t-1}$  的充分条件能满足的情形下, RemoveOnly 将被触发.

(3) 若第  $t-l$  轮迭代中 RemoveOnly 已被触发且  $|L^t| > |L_{de}^{t-1}|$ , 则定理 4 使  $\xi^t < \xi_{de}^{t-1}$  的充分条件和定理 2 使  $\xi_{de}^t < \xi^t$  的充分条件都能满足的情形下(保证  $\xi_{de}^t < \xi^t < \xi_{de}^{t-1}$ ), RemoveOnly 将被触发.

(4) 若第  $t-l$  轮迭代中 RemoveOnly 已被触发且  $|L^t| > |L_{de}^{t-1}|$ , 则在定理 4 使  $\xi^t < \xi_{de}^{t-1}$  的充分条件无法满足, 但定理 5 使  $\xi_{de}^t < \xi_{de}^{t-1}$  的充分条件能满足的情形下, RemoveOnly 将被触发.

(5) 除以上情形外, RemoveOnly 将被抑制.

不难发现, 自适应剪辑策略中的(1)和(3)表明: 在标准 Tri-training 基本训练过程能保证所得假设准确率迭代提高的基础上, ADE-Tri-training 将尽可能触发 RemoveOnly 剪辑操作使准确率最大限度提高, 同时(2)和(4)表明: 在标准 Tri-training 无法保证所得假设准确率迭代提高的时候, ADE-Tri-training 将试图触发剪辑操作使准确率仍能迭代提高. 因此在 RemoveOnly 被触发的 ADE-Tri-training 迭代中所得假设的泛化能力提高幅度将优于标准 Tri-training.

## 4 ADE-Tri-training 算法

ADE-Tri-training 每轮迭代基本过程是: 首先按标准 Tri-training 的互标记方式获得新标记样例集, 并尝试用 RemoveOnly 剪辑新标记样例集, 然后计算 RemoveOnly 两个性能指标并根据当前情形由自适应策略中(1)~(5)决定 RemoveOnly 被触发或

抑制, 若被触发则用剪辑后的新标记样例集更新训练集重新训练; 若被抑制则仍按标准 Tri-training 方式更新训练集重新训练.

算法 1 是 ADE-Tri-training 算法伪码. 为保证由初始训练集  $L$  训练得到 3 个不同的基分类器, 新算法采用 *Bootstrap* 采样技术.  $MeasureError(h_j$  和  $h_k)$  和  $MeasureRecall(h_j$  和  $h_k)$  分别估计联合分类器  $h_j$  和  $h_k$  的分类错误率和 RemoveOnly 对误标记样例的召回率. 由于对无标记样例估计分类错误率和召回率十分困难, 基于无标记样例集  $U$  和初始带标记样例集  $L$  具有相同分布的假设, 在  $L$  上估计. 并用  $L$  中被  $h_j$  和  $h_k$  标记一致的子集  $L'$  上误标记率来估计  $U$  中一致标记子集  $L'$  上误标记率  $\bar{e}'_1$ , 用 RemoveOnly 在  $L'$  上的召回率来估计  $L'$  上召回率  $r'$ .  $RemoveOnly(L_i)$  识别新标记样例集  $L_i$  中误标记样例并移除.

如 3.2 节所述, 判定定理 1 充分条件成立时, 需要判定  $0 < \frac{\bar{e}'_1}{\bar{e}_1^{t-1}} < 1$  和  $0 < \frac{|L^{t-1}|}{|L^t|} < 1$  及  $\bar{e}'_1 |L^t| < \bar{e}_1^{t-1} |L^{t-1}|$  同时成立, 对迭代过程中可能出现的由于  $|L^t|$  远大于  $|L^{t-1}|$  而导致  $0 < \frac{\bar{e}'_1}{\bar{e}_1^{t-1}} < 1$  和  $0 < \frac{|L^{t-1}|}{|L^t|} < 1$  成立但  $\bar{e}'_1 |L^t| < \bar{e}_1^{t-1} |L^{t-1}|$  不成立的情形, ADE-Tri-training 沿用文献[12]的二次采样法: 在  $|L^{t-1}|$  满足  $|L^{t-1}| > \frac{\bar{e}'_1}{\bar{e}_1^{t-1} - \bar{e}'_1}$  前提下(保证  $0 < \frac{|L^{t-1}|}{|L^t|} < 1$  成立), 对  $L^t$  进行随机二次采样, 使  $|L^t| = \left\lceil \frac{\bar{e}_1^{t-1} |L^{t-1}|}{\bar{e}'_1} - 1 \right\rceil$ , 则  $\bar{e}'_1 |L^t| < \bar{e}_1^{t-1} |L^{t-1}|$  也将同时成立. 同样, 判断定理 3~定理 5 的充分条件时, 对可能因  $|L^t|$  远大于  $|L^{t-1}|$  而导致条件不成立的情形, 也根据  $|L^{t-1}|$  大小决定是否对  $L^t$  二次采样使条件满足. 定理 3 中, 若  $|L^{t-1}| > (1-r_i)e_i / (e'_i - (1-r_i)e_i)$ , 则按规模  $\lceil e'_i l'_i / ((1-r_i)e_i) - 1 \rceil$  对  $L^t$  二次采样; 定理 4 中, 若  $|L^{t-1}| > e_i / ((1-r'_i)e'_i - e_i)$ , 则按规模  $\lceil (1-r'_i)e'_i l'_i / e_i - 1 \rceil$  对  $L^t$  二次采样; 定理 5 中, 若  $|L^{t-1}| > (1-r_i)e_i / ((1-r'_i)e'_i - (1-r_i)e_i)$ , 则按规模  $\lceil (1-r'_i)e'_i l'_i / ((1-r_i)e_i) - 1 \rceil$  对  $L^t$  二次采样.

最终假设  $h(x)$  是 3 个分类器的加权投票结果, 权重由初始带标记样例集  $L$  上准确率  $A_i(L)$  决定.

**算法 1.** ADE-Tri-training 算法伪码.

输入: 初始带标记样例集  $L$ ; 无标记样例集  $U$ ; 监督学

习算法 *Learn*

输出：假设  $h(x) = \arg \max_{y \in \text{label}} \frac{\sum_{i=1}^3 \delta(y, h_i(x)) \times A_i(L)}{\sum_{i=1}^3 A_i(L)}$ ，其

中  $\delta(y, h_i(x)) = \begin{cases} 1, & h_i(x) = y \\ 0, & h_i(x) \neq y \end{cases}$

1. for  $i \in \{1..3\}$  do //初始化 3 个不同的基分类器
2.  $S_i \leftarrow \text{BootstrapSample}(L)$
3.  $h_i \leftarrow \text{Learn}(S_i)$
4.  $e'_i \leftarrow 0.5$ ;  $l'_i \leftarrow 0$ ;  $lde'_i \leftarrow 0$ ;  $r'_i \leftarrow 0$ ;  $edit'_i \leftarrow \text{FALSE}$
5. end of for
6. repeat until none of  $h_i (i \in \{1..3\})$  changes
7. for  $i \in \{1..3\}$  do
8.  $L_i \leftarrow \emptyset$ ;  $update_i \leftarrow \text{FALSE}$ ;  $LDE_i \leftarrow \emptyset$ ;  
 $edit_i \leftarrow \text{FALSE}$
9.  $e_i \leftarrow \text{MeasureError}(h_j \text{ 和 } h_k) (j, k \neq i)$
10.  $r_i \leftarrow \text{MeasureRecall}(h_j \text{ 和 } h_k) (j, k \neq i)$
11. for every  $x \in U$  do
12. if  $h_j(x) = h_k(x) (j, k \neq i)$  then  
 $L_i \leftarrow L_i \cup \{(x, h_j(x))\}$
13. end of for
14. if  $edit'_i = \text{FALSE}$  and  $|L_i| > l'_i$  then  
//上一轮剪辑操作未触发
15. if  $e_i < e'_i$  then //判定自适应策略(1)触发情形
16. if  $l'_i = 0$  then //  $h_i$  尚未重新训练过
17.  $l'_i \leftarrow \lfloor e_i / (e'_i - e_i) + 1 \rfloor$
18. if  $e_i |L_i| < e'_i l'_i$  then //判定定理 1 充分条件
19.  $update_i \leftarrow \text{TRUE}$
20. else if  $l'_i > e_i / (e'_i - e_i)$  then //是否需要二次采样
21.  $L_i \leftarrow \text{Subsample}(L_i, \lceil e'_i l'_i / e_i - 1 \rceil)$ ;
22.  $update_i \leftarrow \text{TRUE}$
23. if  $update_i \leftarrow \text{TRUE}$  then //判定定理 2 充分条件
24.  $LDE_i \leftarrow \text{RemoveOnly}(L_i)$
25. if  $r_i \geq (|L_i| - |LDE_i|) / (2e_i |L_i|)$  then
26.  $edit_i \leftarrow \text{TRUE}$
27. else if  $(1 - r_i)e_i < e'_i$  then  
//判定自适应策略(2)触发情形
28. if  $l'_i = 0$  then
29.  $l'_i \leftarrow \lfloor (1 - r_i)e_i / (e'_i - (1 - r_i)e_i) + 1 \rfloor$
30. if  $(1 - r_i)e_i |L_i| < e'_i l'_i$  then  
//判定定理 3 充分条件
31.  $LDE_i \leftarrow \text{RemoveOnly}(L_i)$
32. if  $|LDE_i| > l'_i$  then  $edit_i \leftarrow \text{TRUE}$
33. else //是否需要二次采样
34. if  $l'_i > (1 - r_i)e_i / (e'_i - (1 - r_i)e_i)$  then  
 $L_i \leftarrow \text{Subsample}(L_i,$   
 $\lceil e'_i l'_i / ((1 - r_i)e_i) - 1 \rceil)$
35.  $LDE_i \leftarrow \text{RemoveOnly}(L_i)$

36. if  $|LDE_i| > l'_i$  then  $edit_i \leftarrow \text{TRUE}$
37. if  $edit'_i = \text{TRUE}$  and  $|L_i| > l'_i$  then  
//上一轮剪辑操作被触发
38. if  $e_i < (1 - r'_i)e'_i$  then  
//判定自适应策略(3)触发情形
39. if  $e_i |L_i| < (1 - r'_i)e'_i l'_i$  then  
//判定定理 4 充分条件
40.  $update_i \leftarrow \text{TRUE}$
41. else if  $l'_i > e_i / ((1 - r'_i)e'_i - e_i)$  then  
//是否需二次采样
42.  $L_i \leftarrow \text{Subsample}(L_i, \lceil (1 - r'_i)e'_i l'_i / e_i - 1 \rceil)$
43.  $update_i \leftarrow \text{TRUE}$
44. if  $update_i \leftarrow \text{TRUE}$  then  
//判定定理 2 充分条件
45.  $LDE_i \leftarrow \text{RemoveOnly}(L_i)$
46. if  $r_i \geq (|L_i| - |LDE_i|) / (2e_i |L_i|)$  then
47.  $edit_i \leftarrow \text{TRUE}$
48. else if  $(1 - r_i)e_i < (1 - r'_i)e'_i$  then  
//判定自适应策略(4)
49. if  $(1 - r_i)e_i |L_i| < (1 - r'_i)e'_i l'_i$  then  
//判定定理 5
50.  $LDE_i \leftarrow \text{RemoveOnly}(L_i)$
51. if  $|LDE_i| > lde'_i$  then  $edit_i \leftarrow \text{TRUE}$
52. else //是否需要二次采样
53. if  $l'_i > (1 - r_i)e_i / ((1 - r'_i)e'_i - (1 - r_i)e_i)$   
then  
 $L_i \leftarrow \text{Subsample}(L_i,$   
 $\lceil (1 - r'_i)e'_i l'_i / ((1 - r_i)e_i) - 1 \rceil)$
54.  $LDE_i \leftarrow \text{RemoveOnly}(L_i)$
55. if  $|LDE_i| > l'_i$  then  $edit_i \leftarrow \text{TRUE}$
56. end of for
57. for  $i \in \{1..3\}$  do //重新训练
58. if  $edit_i = \text{TRUE}$  then
59.  $h_i \leftarrow \text{Learn}(L \cup LDE_i)$   
 $e'_i \leftarrow e_i$ ;  $l'_i \leftarrow |L_i|$ ;  $edit'_i \leftarrow edit_i$ ;  $r'_i \leftarrow r_i$ ;  
 $lde'_i \leftarrow |LDE_i|$
60. else if  $update_i = \text{TRUE}$  then  
 $h_i \leftarrow \text{Learn}(L \cup L_i)$   
 $e'_i \leftarrow e_i$ ;  $l'_i \leftarrow |L_i|$ ;  $edit'_i \leftarrow \text{FALSE}$
61. end of for
62. end of repeat

## 5 实验<sup>①</sup>

我们选用文献[9]所用的 12 个 UCI 数据集<sup>[20]</sup>

① 本文实验所用数据集、程序、实验过程详细记录文档均可从 <http://nclab.hit.edu.cn/dc.html> 获得。

进行实验,表 1 列出了这些数据集的信息. 对每个数据集,取出 25% 作测试集,其余的作训练集并按照不同无标记比例(80%、60%、40%、20%)划分为无标记样例集  $U$  和带标记样例集  $L$ . 原则上,测试集、样例集  $L$ 、样例集  $U$  中正反例比率与原始数据集相同.

表 1 实验所用 UCI 数据集

数据集	属性	样本数目	类别	正例/反例 /%
australian	14	690	2	55.5/44.5
bupa	6	345	2	42.0/58.0
colic	22	368	2	63.0/37.0
diabetes	8	768	2	65.1/34.9
german	20	1000	2	70.0/30.0
hypothyroid	25	3163	2	4.8/95.2
ionosphere	34	351	2	35.9/64.1
kr-vs-kp	36	3196	2	52.2/47.8
sick	29	3772	2	6.1/93.9
tic-tac-toe	9	958	2	65.3/34.7
vote	16	435	2	61.4/38.6
wdbc	30	569	2	37.3/62.7

为得到平均性能,对每个给定的无标记比例(如 80%),产生 3 个不同的  $L$  和  $U$  随机划分,对应每个划分独立运行 1 次算法并考察所学假设对测试集的分类错误率,取 3 次独立运行的分类错误率平均值作为算法在该无标记比例(80%)下的错误率.

为比较验证,我们运行 ADE-Tri-training、机械执行 RemoveOnly 剪辑操作的 DE-Tri-training、Tri-training 3 种算法. 其中,训练基分类器的监督学习算法先后采用 Weka<sup>[21]</sup> 提供的 J4.8 决策树、BP 神经网络及朴素贝叶斯方法. 样例间距离计算采用 HVDM<sup>[22]</sup> (Heterogeneous Value Difference Metric).

表 2~表 5 是不同无标记比例下,3 种算法性能统计比较. 其中初始错误率(initial)是第 0 次迭代由初始带标记样例集  $L$  训练所得 3 个初始基分类器决定的联合假设分类错误率,最终错误率(final)是迭代训练结束后联合假设的分类错误率,性能提高百分比(improve)是最终错误率较初始错误率的降低比例. 表中性能提高百分比的最大值用粗体标记(注意表中显示的错误率值是四舍五入的结果).

由表 2~表 5 计算全部无标记比例下 3 种算法在所有数据集上性能提高百分比(即表中全部 144 项 improve 值)的平均值,可以发现 ADE-Tri-training 性能提高最显著为 10.1%, DE-Tri-training 和 Tri-training 分别提高 6.1% 和 6.6%. 其中,无标记比例为 20% 时(表 5)性能提高百分比(36 项)的平均值,DE-Tri-training 仅为 4.3%,低于 Tri-training 的 6.8%,而 ADE-Tri-training 为 10.3%,远高于 Tri-training,这不仅与我们在文献[16]中结论一致,即带标记样例充分时,机械执行剪辑操作将使 Tri-training 泛化性能下降,同时也证明了本文自适应剪辑策略对提高 Tri-training 泛化性能显著有效(尤其采用 BPNN 时这种对比更明显).

由表 2~表 5 统计各种无标记比例下每个算法的胜出次数(获得最大性能提高百分比次数),同样表明 ADE-Tri-training 几乎总是获得最多的胜出次数. 唯有在无标记样例占 60% 且采用朴素贝叶斯时才出现 DE-Tri-training 以 7 次胜出略大于 ADE-Tri-training 的 6 次胜出.

表 2 3 种算法分类错误率比较 1(无标记比例=80%)

(a) 采用 J4.8 决策树方法时的分类错误率

数据集	分类错误率						
	initial	ADE-Tri-training		DE-Tri-training		Tri-training	
		final	improve/%	final	improve/%	final	improve/%
australian	0.202	0.171	<b>14.7</b>	0.194	1.3	0.178	9.6
bupa	0.418	0.391	<b>4.6</b>	0.418	-1.3	0.406	1.8
colic	0.203	0.174	<b>14.1</b>	0.188	6.5	0.181	9.6
diabetes	0.313	0.266	<b>14.0</b>	0.280	9.7	0.309	0.1
german	0.333	0.319	4.3	0.312	<b>6.0</b>	0.323	3.0
hypothyroid	0.011	0.009	<b>19.9</b>	0.009	16.9	0.009	16.9
ionosphere	0.136	0.114	<b>9.7</b>	0.136	-12.5	0.114	<b>9.7</b>
kr-vs-kp	0.036	0.029	<b>16.0</b>	0.033	5.9	0.033	8.1
sick	0.023	0.021	10.2	0.021	<b>12.4</b>	0.022	6.2
tic-tac-toe	0.264	0.236	<b>9.6</b>	0.244	7.9	0.242	9.1
vote	0.043	0.028	21.0	0.028	<b>23.8</b>	0.040	-1.6
wdbc	0.120	0.096	<b>19.8</b>	0.110	9.5	0.101	11.9
ave.	0.175	0.154	<b>13.2</b>	0.164	7.2	0.163	7.0

(b) 采用 BP 神经网络方法时的分类错误率

数据集	分类错误率						
	initial	ADE-Tri-training		DE-Tri-training		Tri-training	
		final	improve/%	final	improve/%	final	improve/%
australian	0.213	0.180	<b>15.4</b>	0.188	11.6	0.200	6.2
bupa	0.372	0.329	<b>9.9</b>	0.341	7.5	0.345	6.9
colic	0.250	0.239	3.6	0.221	<b>11.6</b>	0.239	3.6
diabetes	0.262	0.240	<b>8.6</b>	0.247	6.0	0.252	4.1
german	0.331	0.320	3.2	0.316	<b>4.5</b>	0.328	1.0
hypothyroid	0.034	0.032	5.9	0.030	<b>11.4</b>	0.031	8.5
ionosphere	0.226	0.211	<b>6.9</b>	0.222	1.9	0.222	1.6
kr-vs-kp	0.048	0.047	1.8	0.048	-1.7	0.043	<b>8.3</b>
sick	0.039	0.037	5.1	0.038	2.5	0.034	<b>12.3</b>
tic-tac-toe	0.062	0.053	<b>12.6</b>	0.056	10.2	0.053	<b>12.6</b>
vote	0.052	0.040	<b>23.3</b>	0.046	12.5	0.040	<b>23.3</b>
wdbc	0.068	0.061	<b>10.6</b>	0.066	2.8	0.061	<b>10.6</b>
ave.	0.163	0.149	<b>8.9</b>	0.152	6.7	0.154	8.3

(c) 采用朴素贝叶斯方法时的分类错误率

数据集	分类错误率						
	initial	ADE-Tri-training		DE-Tri-training		Tri-training	
		final	improve/%	final	improve/%	final	improve/%
australian	0.222	0.208	<b>6.0</b>	0.216	2.4	0.218	1.5
bupa	0.477	0.450	<b>5.7</b>	0.453	4.9	0.457	4.0
colic	0.225	0.200	<b>10.8</b>	0.207	7.3	0.203	9.2
diabetes	0.253	0.229	<b>9.3</b>	0.229	9.2	0.231	8.6
german	0.284	0.264	<b>7.2</b>	0.269	5.2	0.271	4.8
hypothyroid	0.019	0.017	<b>7.4</b>	0.019	0.5	0.017	<b>7.4</b>
ionosphere	0.140	0.125	<b>14.6</b>	0.125	13.3	0.129	11.2
kr-vs-kp	0.130	0.121	<b>6.5</b>	0.123	4.6	0.123	4.9
sick	0.074	0.065	<b>10.4</b>	0.069	4.3	0.072	1.4
tic-tac-toe	0.291	0.271	<b>6.9</b>	0.276	4.9	0.278	4.5
vote	0.086	0.080	<b>5.6</b>	0.083	1.9	0.083	1.9
wdbc	0.077	0.065	<b>15.9</b>	0.075	4.1	0.072	6.7
ave.	0.190	0.175	<b>8.9</b>	0.179	5.2	0.179	5.5

表 3 3 种算法分类错误率比较 2(无标记比例=60%)

(a) 采用 J4.8 决策树方法时的分类错误率

数据集	分类错误率						
	initial	ADE-Tri-training		DE-Tri-training		Tri-training	
		final	improve/%	final	improve/%	final	improve/%
australian	0.167	0.157	6.3	0.157	<b>6.6</b>	0.161	3.8
bupa	0.415	0.329	<b>19.5</b>	0.380	7.0	0.372	9.0
colic	0.250	0.199	<b>19.4</b>	0.225	8.5	0.225	8.5
diabetes	0.250	0.224	<b>10.4</b>	0.240	4.1	0.241	3.6
german	0.317	0.299	<b>5.9</b>	0.308	3.1	0.312	1.8
hypothyroid	0.011	0.010	<b>10.0</b>	0.010	<b>10.0</b>	0.011	6.7
ionosphere	0.125	0.110	3.6	0.125	-8.7	0.102	<b>10.7</b>
kr-vs-kp	0.012	0.009	<b>17.6</b>	0.010	13.9	0.009	<b>17.6</b>
sick	0.023	0.017	<b>24.6</b>	0.017	22.2	0.017	23.8
tic-tac-toe	0.208	0.187	<b>9.9</b>	0.205	0.7	0.199	3.5
vote	0.064	0.049	<b>20.8</b>	0.061	4.2	0.052	16.7
wdbc	0.061	0.047	25.1	0.045	<b>27.5</b>	0.049	20.9
ave.	0.159	0.136	<b>14.4</b>	0.148	8.3	0.146	10.5

(b) 采用 BP 神经网络方法时的分类错误率

数据集	分类错误率						
	initial	ADE-Tri-training		DE-Tri-training		Tri-training	
		final	improve/%	final	improve/%	final	improve/%
australian	0.196	0.176	9.9	0.167	<b>15.1</b>	0.174	10.8
bupa	0.384	0.345	<b>9.9</b>	0.357	7.0	0.372	3.0
colic	0.228	0.192	13.1	0.185	<b>16.4</b>	0.199	10.6
diabetes	0.283	0.260	<b>7.0</b>	0.292	-3.7	0.276	1.3
german	0.319	0.301	6.1	0.295	<b>7.8</b>	0.308	3.5
hypothyroid	0.029	0.025	<b>12.0</b>	0.033	-14.3	0.028	0.8
ionosphere	0.161	0.126	<b>21.1</b>	0.138	14.2	0.134	14.9
kr-vs-kp	0.018	0.013	<b>25.0</b>	0.013	24.6	0.014	21.3
sick	0.028	0.026	<b>9.2</b>	0.028	2.6	0.032	-11.6
tic-tac-toe	0.017	0.015	14.8	0.014	17.8	0.014	<b>19.4</b>
vote	0.022	0.012	<b>44.4</b>	0.015	27.8	0.012	<b>44.4</b>
wdbc	0.052	0.042	<b>18.5</b>	0.054	-5.4	0.052	-0.6
ave.	0.145	0.128	<b>15.9</b>	0.132	9.2	0.135	9.8

(c) 采用朴素贝叶斯方法时的分类错误率

数据集	分类错误率						
	initial	ADE-Tri-training		DE-Tri-training		Tri-training	
		final	improve/%	final	improve/%	final	improve/%
australian	0.212	0.208	<b>1.8</b>	0.210	0.9	0.210	0.9
bupa	0.426	0.415	1.6	0.407	<b>3.3</b>	0.407	<b>3.3</b>
colic	0.210	0.207	1.7	0.207	1.7	0.203	<b>3.4</b>
diabetes	0.240	0.234	2.2	0.233	<b>2.9</b>	0.233	<b>2.9</b>
german	0.268	0.249	<b>6.8</b>	0.253	5.3	0.253	5.3
hypothyroid	0.025	0.025	<b>1.7</b>	0.025	<b>1.7</b>	0.025	<b>1.7</b>
ionosphere	0.123	0.115	4.8	0.111	<b>5.8</b>	0.115	3.4
kr-vs-kp	0.130	0.126	<b>3.1</b>	0.126	2.5	0.127	2.2
sick	0.087	0.080	<b>6.5</b>	0.082	3.9	0.083	3.5
tic-tac-toe	0.308	0.297	3.6	0.294	<b>4.5</b>	0.304	1.4
vote	0.086	0.086	0.0	0.083	<b>3.7</b>	0.083	<b>3.7</b>
wdbc	0.052	0.040	<b>21.4</b>	0.040	<b>21.4</b>	0.042	15.9
ave.	0.181	0.174	4.6	0.173	<b>4.8</b>	0.174	4.0

表 4 3 种算法分类错误率比较 3(无标记比例=40%)

(a) 采用 J4.8 决策树方法时的分类错误率

数据集	分类错误率						
	initial	ADE-Tri-training		DE-Tri-training		Tri-training	
		final	improve/%	final	improve/%	final	improve/%
australian	0.161	0.155	<b>3.5</b>	0.157	1.2	0.155	2.8
bupa	0.395	0.368	6.6	0.375	5.3	0.360	<b>8.7</b>
colic	0.163	0.159	-0.4	0.159	-0.2	0.156	<b>1.7</b>
diabetes	0.311	0.273	<b>11.6</b>	0.286	7.3	0.306	1.7
german	0.311	0.281	<b>9.1</b>	0.281	8.9	0.301	2.7
hypothyroid	0.013	0.011	<b>12.2</b>	0.011	9.4	0.011	10.8
ionosphere	0.114	0.091	<b>18.6</b>	0.102	8.9	0.106	5.3
kr-vs-kp	0.008	0.008	3.2	0.008	3.2	0.007	<b>12.7</b>
sick	0.018	0.016	<b>8.7</b>	0.016	8.2	0.016	5.8
tic-tac-toe	0.192	0.172	<b>10.0</b>	0.192	-1.1	0.181	4.1
vote	0.059	0.043	<b>18.5</b>	0.049	11.1	0.056	3.7
wdbc	0.087	0.070	<b>16.6</b>	0.085	1.6	0.073	14.4
ave.	0.152	0.137	<b>9.9</b>	0.144	5.3	0.144	6.2

(b) 采用 BP 神经网络方法时的分类错误率

数据集	分类错误率						
	initial	ADE-Tri-training		DE-Tri-training		Tri-training	
		final	improve/%	final	improve/%	final	improve/%
australian	0.176	0.161	<b>8.4</b>	0.171	3.3	0.161	<b>8.4</b>
bupa	0.380	0.357	6.0	0.384	-1.2	0.372	1.8
colic	0.214	0.203	4.3	0.196	<b>8.1</b>	0.203	4.3
diabetes	0.271	0.252	<b>6.9</b>	0.253	6.0	0.260	3.5
german	0.264	0.239	<b>9.7</b>	0.272	-2.8	0.268	-1.5
hypothyroid	0.021	0.020	4.9	0.018	<b>15.0</b>	0.021	4.5
ionosphere	0.192	0.161	<b>15.5</b>	0.165	13.7	0.184	3.5
kr-vs-kp	0.012	0.011	8.7	0.009	<b>22.4</b>	0.011	8.7
sick	0.031	0.025	17.8	0.025	<b>20.2</b>	0.029	7.1
tic-tac-toe	0.025	0.022	<b>8.4</b>	0.024	3.7	0.024	2.9
vote	0.037	0.034	<b>8.3</b>	0.034	6.7	0.034	3.9
wdbc	0.028	0.021	<b>22.2</b>	0.019	<b>22.2</b>	0.026	-5.6
ave.	0.138	0.125	<b>10.1</b>	0.131	9.8	0.133	3.5

(c) 采用朴素贝叶斯方法时的分类错误率

数据集	分类错误率						
	initial	ADE-Tri-training		DE-Tri-training		Tri-training	
		final	improve/%	final	improve/%	final	improve/%
australian	0.256	0.236	<b>7.5</b>	0.240	6.0	0.244	4.5
bupa	0.471	0.456	3.6	0.448	<b>5.2</b>	0.460	2.7
colic	0.272	0.261	<b>3.9</b>	0.264	2.5	0.264	2.5
diabetes	0.226	0.220	2.4	0.215	<b>4.3</b>	0.215	<b>4.3</b>
german	0.257	0.249	<b>3.2</b>	0.251	2.7	0.249	<b>3.2</b>
hypothyroid	0.023	0.022	<b>3.5</b>	0.023	-0.1	0.023	1.8
ionosphere	0.180	0.176	<b>1.9</b>	0.176	<b>1.9</b>	0.176	<b>1.9</b>
kr-vs-kp	0.147	0.141	<b>4.6</b>	0.141	4.3	0.142	3.4
sick	0.076	0.067	<b>11.0</b>	0.067	10.5	0.068	10.0
tic-tac-toe	0.260	0.250	<b>3.7</b>	0.253	2.7	0.253	2.7
vote	0.090	0.083	<b>6.7</b>	0.083	<b>6.7</b>	0.083	<b>6.7</b>
wdbc	0.110	0.108	<b>2.2</b>	0.110	0.0	0.108	<b>2.2</b>
ave.	0.197	0.189	<b>4.5</b>	0.189	3.9	0.190	3.8

表 5 3 种算法分类错误率比较 4(无标记比例=20%)

(a) 采用 J4.8 决策树方法时的分类错误率

数据集	分类错误率						
	initial	ADE-Tri-training		DE-Tri-training		Tri-training	
		final	improve/%	final	improve/%	final	improve/%
australian	0.162	0.143	<b>11.1</b>	0.148	7.8	0.146	9.2
bupa	0.326	0.302	<b>5.8</b>	0.333	-4.7	0.326	-1.8
colic	0.152	0.112	<b>24.4</b>	0.138	8.9	0.138	8.9
diabetes	0.262	0.245	<b>6.5</b>	0.255	2.8	0.253	3.3
german	0.292	0.283	<b>1.8</b>	0.303	-5.1	0.287	1.0
hypothyroid	0.011	0.008	<b>18.5</b>	0.010	7.4	0.010	7.4
ionosphere	0.140	0.125	10.5	0.102	<b>26.3</b>	0.133	5.3
kr-vs-kp	0.007	0.007	<b>4.4</b>	0.008	-8.9	0.007	<b>4.4</b>
sick	0.014	0.013	<b>9.8</b>	0.015	-3.8	0.014	-1.8
tic-tac-toe	0.141	0.124	<b>8.4</b>	0.141	-3.9	0.126	7.5
vote	0.059	0.040	<b>28.1</b>	0.040	25.2	0.040	25.2
wdbc	0.068	0.056	<b>22.1</b>	0.068	-1.7	0.058	19.0
ave.	0.136	0.122	<b>12.6</b>	0.130	4.2	0.128	7.3

(b) 采用 BP 神经网络方法时的分类错误率

数据集	分类错误率						
	initial	ADE-Tri-training		DE-Tri-training		Tri-training	
		final	improve/%	final	improve/%	final	improve/%
australian	0.145	0.136	<b>6.4</b>	0.141	2.2	0.143	0.9
bupa	0.345	0.329	<b>4.4</b>	0.357	-3.7	0.341	0.8
colic	0.178	0.149	<b>16.4</b>	0.156	11.4	0.163	7.6
diabetes	0.276	0.243	<b>11.6</b>	0.252	8.6	0.262	4.9
german	0.289	0.271	<b>6.4</b>	0.280	3.1	0.275	5.1
hypothyroid	0.020	0.018	10.6	0.021	-2.0	0.017	<b>16.6</b>
ionosphere	0.106	0.061	38.9	0.057	<b>44.4</b>	0.061	38.9
kr-vs-kp	0.012	0.010	<b>16.5</b>	0.010	9.6	0.010	<b>16.5</b>
sick	0.028	0.025	<b>10.4</b>	0.027	2.3	0.027	3.7
tic-tac-toe	0.029	0.026	<b>8.3</b>	0.033	-17.5	0.026	<b>8.3</b>
vote	0.034	0.031	<b>5.6</b>	0.037	-11.1	0.031	<b>5.6</b>
wdbc	0.026	0.021	<b>16.7</b>	0.026	2.8	0.023	2.8
ave.	0.124	0.110	<b>12.7</b>	0.116	4.2	0.115	9.3

(c) 采用朴素贝叶斯方法时的分类错误率

数据集	分类错误率						
	initial	ADE-Tri-training		DE-Tri-training		Tri-training	
		final	improve/%	final	improve/%	final	improve/%
australian	0.231	0.229	<b>0.8</b>	0.231	0.0	0.231	0.0
bupa	0.403	0.360	<b>11.2</b>	0.376	7.4	0.388	4.6
colic	0.228	0.214	<b>6.4</b>	0.217	4.7	0.217	4.7
diabetes	0.243	0.236	<b>2.9</b>	0.238	2.1	0.238	2.1
german	0.267	0.259	<b>3.1</b>	0.260	2.6	0.260	2.6
hypothyroid	0.014	0.012	<b>11.6</b>	0.012	<b>11.6</b>	0.013	8.6
ionosphere	0.189	0.170	10.4	0.167	<b>12.6</b>	0.170	10.4
kr-vs-kp	0.136	0.129	<b>5.3</b>	0.130	4.7	0.130	4.7
sick	0.059	0.055	<b>5.6</b>	0.057	3.2	0.057	2.6
tic-tac-toe	0.317	0.310	<b>2.2</b>	0.314	0.9	0.314	0.9
vote	0.077	0.074	<b>3.7</b>	0.077	0.0	0.077	0.0
wdbc	0.063	0.061	3.7	0.061	<b>4.2</b>	0.061	<b>4.2</b>
ave.	0.186	0.176	<b>5.6</b>	0.178	4.5	0.180	3.8

图 1~图 3 分别给出基分类器采用 J4.8、BP 神经网络、朴素贝叶斯训练时,不同无标记比例下 3 种算法在所有数据集上平均错误率随训练过程的迭代变化情况.需要指出的是每个算法每次训练结束的迭代次数都不相同,为便于比较错误率变化过程,图中以 3 种算法训练结束时所有不同迭代次数中最大值为横轴迭代次数最大值,并以每次训练结束时相应错误率值作为其后续迭代次数的值保持不变直至

横轴的最大迭代次数.图中 single 水平线是仅由  $L$  训练所得单个分类器的错误率的平均值.

结合图 1~图 3 考察 3 种算法的计算量,可看出采用不同基分类器在不同无标记比例下 3 种算法的最大迭代次数均不超过 5 次,且第 3 次迭代后平均错误率不再显著变化,尤其采用朴素贝叶斯时第 2 次迭代后已基本不再变化.因此从训练结束时迭代次数角度分析 3 种算法计算量差别不大.

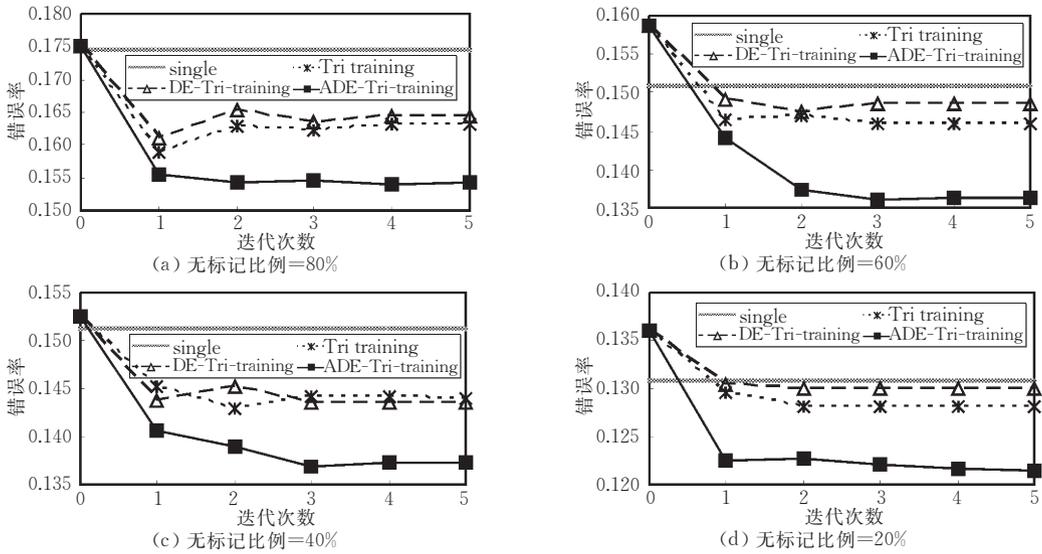


图 1 采用 J4.8 训练基分类器时,不同无标记比例下,3 种算法的错误率平均值迭代变化过程

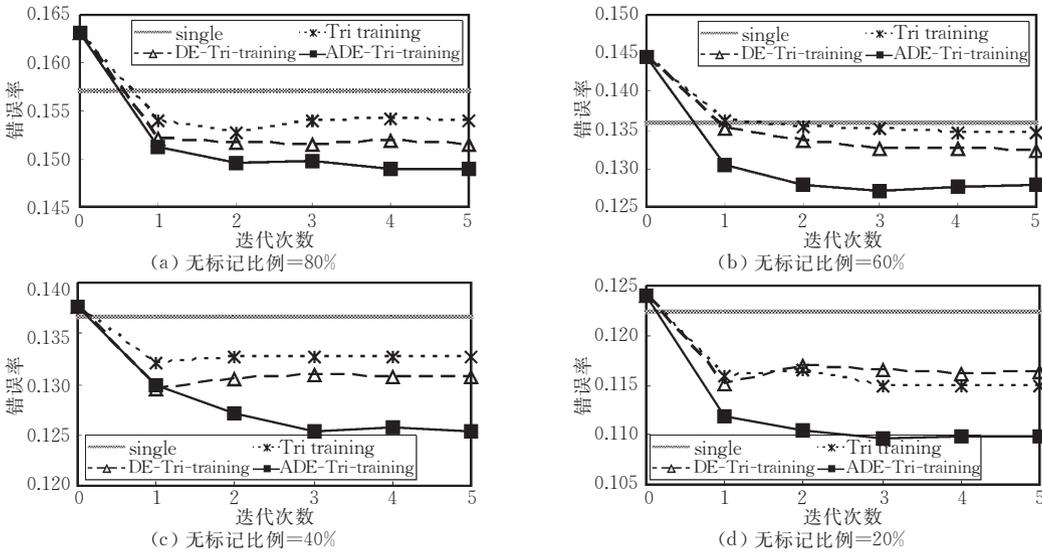


图 2 采用 BP 神经网络训练基分类器时,不同无标记比例下,3 种算法的错误率平均值迭代变化过程

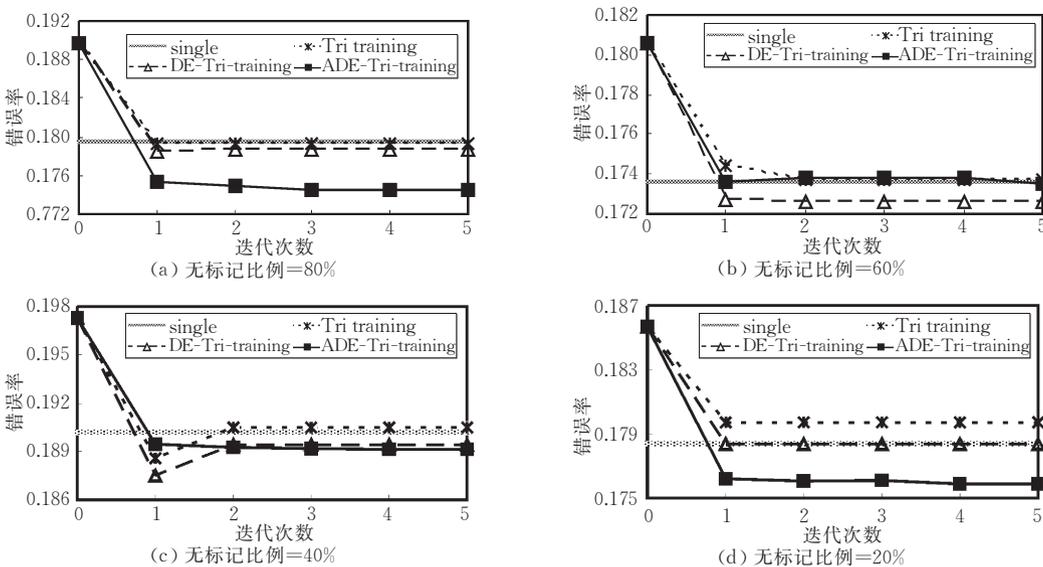


图 3 采用朴素贝叶斯训练基分类器时,不同无标记比例下,3 种算法的错误率平均值迭代变化过程

图 1~图 3 显示,所有情形下,三种算法所得最终假设分类错误率都低于初始假设,这说明三种算法都能利用无标记样例提高泛化性能.同时,易看出 ADE-Tri-training 对泛化性能提高最显著: ADE-Tri-training 不仅几乎每次迭代性能都明显优于 Tri-training,而且在绝大多数情况下也明显优于 DE-Tri-training.此外,对比算法各自迭代过程,发现 ADE-Tri-training 平均错误率基本是持续降低直至迭代结束达最低值,几乎不会出现 Tri-training 和 DE-Tri-training 的波动甚至上扬现象,这也直观表明 ADE-Tri-training 每次迭代受误标记噪声影响最小,能更稳定地保证泛化性能的提高,有更好的健壮性.

## 6 结 论

针对 Tri-training 迭代训练过程中互标记引起的噪声问题,本文提出了 ADE-Tri-training 算法.在引入数据剪辑技术消除互标记噪声的原有工作基础上,对剪辑操作的性能定量刻画,证明不同迭代情形下剪辑操作应被触发的充分条件,形成由触发条件组成的自适应剪辑策略, ADE-Tri-training 用自适应剪辑策略扩展并代替 Tri-training 算法的训练集迭代更新准则,保证了每次迭代时剪辑操作的触发会使泛化能力提高更大. UCI 数据集上大量实验表明 ADE-Tri-training 泛化性能和性能迭代提高的稳定性较 Tri-training 和 DE-Tri-training 都有显著提高.

由于 RemoveOnly 剪辑操作和自适应剪辑策略不仅能有效去除互标记噪声而且克服了机械触发剪辑操作的弊端,所以本文自适应数据剪辑策略的思想对结合数据剪辑技术到其它半监督学习算法的研究工作同样具有借鉴意义.

## 参 考 文 献

- [1] Chapelle O, Schoelkopf B, Zien A. *Semi-Supervised Learning*. Cambridge: MIT Press, 2006
- [2] Nigam K, McCallum A K, Thrun S, Mitchell T. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 2000, 39(2-3): 103-134
- [3] Miller D J, Browning J. A mixture model and EM-based algorithm for class discovery, robust classification, and outlier rejection in mixed labeled/unlabeled data sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003, 25(11): 1468-1483
- [4] Joachims T. Transductive inference for text classification using support vector machines//*Proceedings of the 16th International Conference on Machine Learning*. New York, USA, 1999: 200-209
- [5] Blum A, Lafferty J, Rwebangira M, Reddy R. Semi-supervised learning using randomized mincuts//*Proceedings of the 21st International Conference on Machine Learning*. Texas, USA, 2004: 934-947
- [6] Zhu X J. *Semi-supervised learning literature survey*. University of Wisconsin, Wisconsin: Technical Report: TR1530, 2006
- [7] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training//*Proceedings of the 11th annual conference on Computational Learning Theory*. Wisconsin, USA, 1998: 92-100
- [8] Goldman S, Zhou Y. Enhancing supervised learning with unlabeled data//*Proceedings of the 17th International Conference on Machine Learning*. California, USA, 2000: 327-334
- [9] Zhou Z H, Li M. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 2005, 17(11): 1529-1541
- [10] Blum A, Chawla S. Learning from labeled and unlabeled data using graph mincuts//*Proceedings of the 18th International Conference on Machine Learning*. Williamstown, MA, 2001: 19-26
- [11] Vincent N, Claire C. Bootstrapping coreference classifiers with multiple machine learning algorithms//*Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. Sapporo, Japan, 2003: 113-120
- [12] Hwa R, Osborne M, Sarkar A, Steedman M. Corrected co-training for statistical parsers//*Proceedings of the ICML03 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*. Washington, DC, 2003: 95-102
- [13] Li M, Zhou Z H. SETRED: Self-training with editing//*Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Hanoi, Vietnam, 2005: 611-621
- [14] Nigam K, Ghani R. Analyzing the effectiveness and applicability of co-training//*Proceedings of the ACM 9th Conference on Information and Knowledge Management*. Washington, DC, 2000: 86-93
- [15] Muhlenbach F, Lallich S, Zighed D A. Identifying and handling mislabeled instances. *Journal of Intelligent Information Systems*, 2004, 22(1): 89-109
- [16] Deng C, Guo M Z. Tri-training and data editing based semi-supervised clustering algorithm//*Proceedings of the 5th Mexican International Conference on Artificial Intelligence*. Mexico, 2006: 641-651
- [17] Jiang Y, Zhou Z H. Editing training data for kNN classifiers with neural network ensemble//*Proceedings of the 1st International Symposium on Neural Networks*. Dalian, China, 2004: 356-361

- [18] Sánchez J S, Barandela R, Marqués A I, Alejo R, Badenas J. Analysis of new techniques to obtain quality training sets. *Pattern Recognition Letters*, 2003, 24(7): 1015-1022
- [19] Angluin D, Laird P. Learning from noisy examples. *Machine Learning*, 1988, 2(4): 343-370
- [20] Blake C, Keogh E, Merz C J. UCI repository of machine learning databases. <http://www.ics.uci.edu/~mlearn/>

MLRepository.html

- [21] Witten I H, Frank E. *Data Mining: Practical Machine Learning Tools and Techniques*. 2nd Edition. San Fransisco: Morgan Kaufmann, 2005
- [22] Wilson D R, Martinez T R. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 1997, 6: 1-34



**DENG Chao**, born in 1978, Ph. D. candidate. His main research interests include machine learning, data mining and pattern recognition.

**GUO Mao-Zu**, born in 1966, Ph. D. , professor, Ph. D. supervisor. His main research interests include machine learning, data mining, computational biology and bioinformatics.

## Background

Traditional supervised learning use only labeled examples to train. However, labeled instances are often difficult, expensive, or time consuming to obtain, as they require the efforts of experienced human annotators. Meanwhile unlabeled data may be relatively easy to collect, especially in Bioinformatics. Therefore, besides Bioinformatics, many other data mining tasks turn to a new machine learning mode named semi-supervised learning that exploits large number of unlabeled examples and little labeled examples to improve classifier's generalization ability. Co-training is a well-known semi-supervised classification model, and Tri-training is a revised Co-training style semi-supervised algorithm. Compared with standard Co-training and other revised versions, Tri-training has many advantages due to employing three base-classifiers. However, Tri-training still suffers from the common disadvantage in Co-training style algorithms, i. e. the performance is not stable due to the unlabeled examples may often be wrongly labeled and accumulated during the learning process. There are very few efforts to solve the problem in most current research works.

The objective of this work is to detect and remove the wrongly labeled examples during labeling unlabeled examples so that improve the stability and generalization ability of Tri-training. This approach provides a revised type of Tri-training, ADE-Tri-training. Compared with Tri-training, ADE-Tri-training employs RemoveOnly data editing operation and

adopts adaptive strategy to clean newly labeled examples and control the trigger or inhibit of RemoveOnly operation. Thus the labeling process could adaptively reduce the wrongly labeled examples according to different situations and effectively avoid the negative effect of RemoveOnly. This algorithm can be used in many tasks, such as bioinformatics, text mining, web page and image classification.

This work is mainly supported by the National Natural Science Foundation of China under grant No. 60671011 (Research on Class-Driven RNA Secondary Structure Prediction Algorithms) and the Science Fund for Distinguished Young Scholars of Heilongjiang Province in China under grant No. JC200611 (Research on Machine Learning Algorithms for Computational Biology). These projects are focused on developing effective machine learning approaches for biological data processing and modeling. The research group has done some efforts such as proposed a permutation and GA based RNA secondary structure prediction practical approach, a PSO and EM based phylogenetic tree construction approach, etc. Prior to the work in this paper, they have proposed Tri-training and data editing based semi-supervised clustering algorithms, DE-Tri-training based K-means, in which they observed the disadvantage of trigger data editing operation by rote. Furthermore, they address it through works in this paper.