

多标签代价敏感分类集成学习算法

付忠良¹

摘 要 尽管多标签分类问题可以转换成一般多分类问题解决, 但多标签代价敏感分类问题却很难转换成多类代价敏感分类问题. 通过对多分类代价敏感学习算法扩展为多标签代价敏感学习算法时遇到的一些问题进行分析, 提出了一种多标签代价敏感分类集成学习算法. 算法的平均错分代价为误检标签代价和漏检标签代价之和, 算法的流程类似于自适应提升 (Adaptive boosting, AdaBoost) 算法, 其可以自动学习多个弱分类器来组合成强分类器, 强分类器的平均错分代价将随着弱分类器增加而逐渐降低. 详细分析了多标签代价敏感分类集成学习算法和多类代价敏感 AdaBoost 算法的区别, 包括输出标签的依据和错分代价的含义. 不同于通常的多类代价敏感分类问题, 多标签代价敏感分类问题的错分代价要受到一定的限制, 详细分析并给出了具体的限制条件. 简化该算法得到了一种多标签 AdaBoost 算法和一种多类代价敏感 AdaBoost 算法. 理论分析和实验结果均表明提出的多标签代价敏感分类集成学习算法是有效的, 该算法能实现平均错分代价的最小化. 特别地, 对于不同类错分代价相差较大的多分类问题, 该算法的效果明显好于已有的多类代价敏感 AdaBoost 算法.

关键词 多标签分类, 代价敏感学习, 集成学习, 自适应提升算法, 多分类

引用格式 付忠良. 多标签代价敏感分类集成学习算法. 自动化学报, 2014, 40(6): 1075–1085

DOI 10.3724/SP.J.1004.2014.01075

Cost-sensitive Ensemble Learning Algorithm for Multi-label Classification Problems

FU Zhong-Liang¹

Abstract Although a multi-label classification problem can be converted into a multi-class classification problem to solve, it is difficult that a multi-label cost-sensitive classification problem is converted into a multi-class cost-sensitive classification problem. A cost-sensitive ensemble learning algorithm for multi-label classification problems is proposed based on the analysis on the problems encountered when the multi-class cost-sensitive learning algorithm being extended to multi-label cost-sensitive learning algorithms. The average misclassification cost of the algorithm is composed of fall-out cost and the omission cost. The new algorithm's process is similar to the adaptive boosting (AdaBoost) algorithm, and the algorithm can automatically learn some weak classifiers and combine them into a strong classifier, and the average misclassification cost of the strong classifier will decrease as the weak classifiers gradually increase. The distinction between the cost-sensitive ensemble learning algorithm for multi-label classification problems and the cost-sensitive AdaBoost algorithm for multi-class classification problems is analyzed in detail, including the basis of output label and the meaning of the misclassification cost. Unlike general multi-class cost-sensitive classification problems, the misclassification cost of the multi-label cost-sensitive classification problems are subject to certain restrictions, and the specific restrictions are given. A multi-label AdaBoost algorithm and a multi-class cost-sensitive AdaBoost algorithm can be obtained by simplifying the proposed algorithm. Theoretical analysis and experimental results show that the proposed multi-label cost-sensitive classification ensemble learning algorithm is effective, and that the algorithm can minimize the average misclassification cost. In particular, when the difference of costs of the classes is large, the proposed algorithm can get better results than the existing multi-class cost-sensitive AdaBoost algorithms.

Key words Multi-label classification, cost-sensitive learning, ensemble learning, adaptive boosting (AdaBoost) algorithm, multi-class classification

Citation Fu Zhong-Liang. Cost-sensitive ensemble learning algorithm for multi-label classification problems. *Acta Automatica Sinica*, 2014, 40(6): 1075–1085

收稿日期 2013-07-30 录用日期 2013-09-29
Manuscript received July 30, 2013; accepted September 29, 2013

四川省科技支撑计划 (2011GZ0171, 2012GZ0106) 资助
Supported by the Key Technology Research and Development Program of Sichuan Province of China (2011GZ0171, 2012GZ0106)

本文责任编辑 刘成林
Recommended by Associate Editor LIU Cheng-Lin
1. 中国科学院成都计算机应用研究所 成都 610041

多标签分类是指示例可以同时归属多个类别的分类, 其示例由标签集的一个子集而非类别来标识, 引入错分代价的多标签分类称为多标签代价敏感分类, 而通过学习来构造分类器时, 就引出了代价敏感学习 (Cost-sensitive learning)^[1]. 错分代价完全相

1. Chengdu Computer Applications Institute, Chinese Academy of Sciences, Chengdu 610041

等的代价敏感学习问题将自然简化为普通的学习问题, 因此, 研究多标签代价敏感学习具有更普遍的意义.

现实中有不少多标签代价敏感分类问题. 例如在进行钞票或支票的印刷质量检测时, 不仅要检出废品, 还需要对废品进行分类, 如墨淡、墨浓、缺笔、错码、断线、错位等. 废品可能同时有多种错误, 这就涉及到多标签标识. 生产中, 一般要求不能漏检而宁愿误检, 因为废品因漏检而进入流通将产生严重后果, 而误检仅仅减少成品率并增加人工二次剔废成本, 因此漏检和误检代价不同. 进一步, 不同错误类型产品漏检的代价不一样, 如轻微墨淡或墨浓产品漏检代价较小, 而错码产品漏检代价很大.

目前, 已有不少代价敏感学习算法, 如以 C4.5 算法为基础的代价敏感变体方法 C4.5cs^[2]、用元代价处理把一般的分类模型转换成代价敏感分类模型的方法^[3]、用错分代价调整样本初始分布的算法^[4-5]、最小代价决策分类法^[6-8]等, 文献 [9] 对一些代价敏感分类算法进行了比较. 基于自适应提升 (Adaptive boosting, AdaBoost) 算法引入错分代价的代价敏感学习算法^[10-11] 因其良好特性并只要求构造简单分类器而受到了更多的关注, 文献 [12-14] 提出了一些多分类代价敏感 AdaBoost 算法, 其中多分类代价敏感学习算法 (Cost-sensitive AdaBoost for multi-class problems, Cost-MCPBoost)^[12] 还能使错分示例尽量趋向代价相对较小的类.

然而, 关于多标签代价敏感分类学习算法的研究很少, 究其原因是问题的复杂性. 首先, 多标签分类学习的相关理论还没有完全建立起来, 而构造多标签代价敏感学习算法就更无从着手. 尽管多标签分类算法的研究已取得不少成果^[15-17], 但大都是已有单标签分类算法的扩展或延伸. 如幂集法^[18] 和一对多分解法^[19] 将多标签分类问题转换成更多类的单标签分类问题, 而多标签 K 近邻算法 (Multi-label K -nearest neighbor algorithm, ML-kNN)^[20]、排序支持向量机 (Ranking support vector machine, Rank-SVM)^[21]、最小二乘支持向量机 (Least squares support vector machine, LS-SVM)^[22]、多类多标签 AdaBoost (Multi-class multi-label ranking AdaBoost algorithm, AdaBoost.MH)^[23-25] 算法等, 则都是基于已有单标签分类算法进行扩展而得, 很难直接引入错分代价. 将一般的代价敏感分类算法扩展成多标签代价敏感分类算法时, 需要解决如下 3 个问题:

1) 错分代价的确定. K 类代价敏感分类问题的任一示例属于且仅属于一个类, 其错分代价可用对

角元素为 0 而其他元素大于 0 的代价矩阵来标识各种错分情况, 正确分类时代价为 0. 然而, 多标签分类问题则无法这样来确定错分代价. 在多标签分类问题中, 标签集的任意子集可能是某个示例的标签集, 如果仍然按照分类结果完全正确对应的错分代价为 0, 则将引出一个 2^K 个类别的分类问题. 问题不仅被复杂化, 而且确定错分代价也非常困难, 比如具有部分相同标签的两个示例的错分代价就很难确定, 如果采取累加各单个标签的错分代价, 则又无法区分每单个标签的漏检或误检.

2) 多标签分类问题分类错误的定义. 如果预测标签集完全等于真实标签集才算分类正确, 同样会引出 2^K 个类别的分类问题. 多标签分类错误的定义有很多种^[20], 如海明损失 (Hamming loss)、单一错误 (One-error)、覆盖率 (Coverage)、排序损失 (Ranking loss)、平均精度 (Average precision), 而如何引入合理的错分代价值得分析.

3) 对错分代价的限制问题. 多分类问题要求错分代价大于 0 即可, 但多标签分类问题对错分代价的要求却复杂得多. 如果错分代价不加限制, 输出全标签的分类器可能使平均错分代价最小, 此时学习算法将失效.

如果通过集成学习来训练分类器, 一般还要求平均错分代价随着弱分类器的增加而逐渐降低, 这是集成学习算法重要而必须的要求. 文献 [12] 提到多标签分类集成学习算法并引入了代价, 但其无法满足平均错分代价随着弱分类器的增加而逐渐降低这一重要要求, 理论上无法保证学习算法的有效性.

本文针对上述问题进行了深入分析与研究, 构造出可直接应用于多标签代价敏感分类问题的集成学习算法. 该算法能确保平均错分代价随着弱分类器的增加而逐渐降低. 文中给出了算法的平均错分代价上界估计公式, 详细分析了多标签代价敏感分类问题错分代价的限制条件, 给出了基于标签密度的错分代价限制公式. 当所有代价相等时, 简化本文算法, 可得到一种新的多标签分类集成学习算法, 其与已有连续 AdaBoost (Real AdaBoost) 算法^[25] 具有本质的区别.

1 多分类代价敏感 AdaBoost 算法分析

1.1 多分类代价敏感 AdaBoost 算法简介

样本 $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ 来源于示例空间 X , $y_i \in \{1, 2, \dots, K\}$. 弱分类器 $h_t(x)$ 输出标签 l 的置信度为 $h_t(x, l)$, $l = 1, 2, \dots, K$, $f(x) = \sum_{t=1}^T h_t(x)$, 即 $f(x, l) = \sum_{t=1}^T h_t(x, l)$. $H(x) = \arg \max_l \{f(x, l)\}$ 为组合分类器, 其输出最大置信度对应的标签. $c(i, j)_{K \times K}$

为代价矩阵, $c(i, j)$ 为 i 类错误分成 j 类的代价, $c(i, j) \geq 0$, $c(i, i) = 0$, $i, j = 1, 2, \dots, K$. 于是式 (1) 就是平均错分代价, 集成学习算法通常基于最小化式 (1) 来构造.

$$\varepsilon_{cs} = \sum_{i=1}^m \left(\omega_i \sum_{l=1}^K c(y_i, l) \mathbb{I}[H(x_i) = l] \right) \quad (1)$$

通常情况下取 $\omega_i = 1/m$. 如果 $H(x_i) = k$, 则表明 $f(x_i, k) = \max_l \{f(x_i, l)\}$, 此时 $f(x_i, k) \geq \bar{f}(x_i)$, 于是 $\mathbb{I}[H(x_i) = l] \leq \exp(f(x_i, l) - \bar{f}(x_i))$. 其中 $\bar{f}(x) = \frac{1}{K} \sum_{l=1}^K f(x, l)$, $\bar{h}(x) = \frac{1}{K} \sum_{l=1}^K h(x, l)$, 因此

$$\begin{aligned} \varepsilon_{cs} &\leq \sum_{i=1}^m \left(\omega_i \sum_{l=1}^K c(y_i, l) \exp(f(x_i, l) - \bar{f}(x_i)) \right) = \\ &Z_0 \sum_{i=1}^m \sum_{l=1}^K \left(\omega_{i,l}^1 \prod_{t=1}^T \exp(h_t(x_i, l) - \bar{h}_t(x_i)) \right) \end{aligned} \quad (2)$$

其中, $\omega_{i,l}^1 = (\omega_i c(y_i, l))/Z_0$, Z_0 为 $\omega_{i,l}^1$ 的归一化因子. 多分类代价敏感 AdaBoost 算法^[12] 为:

算法 1. 多分类代价敏感 AdaBoost 算法

- 1) 初始化权值: $\omega_{i,l}^1 = c(y_i, l)/(mZ_0)$, $i = 1, 2, \dots, m$, $l = 1, 2, \dots, K$, Z_0 是归一化因子;
- 2) Do for $t = 1, 2, \dots, T$
 - a) 训练弱分类器:
 - i) 计算 $p_t^{j,l}$: 对应划分 $S = S_1^t \cup S_2^t \cup \dots \cup S_{n_t}^t$, 计算 $p_t^{j,l} = \sum_{i:(x_i \in S_j^t)} \omega_{i,l}^t$, $j = 1, 2, \dots, n_t$;
 - ii) 定义 $h_t(x)$: $\forall x \in S_j^t$, $h_t(x, l) = -\ln(p_t^{j,l})$;
 - iii) 选取 $h_t(x)$:
 - 最小化 $Z_t = K \sum_{j=1}^{n_t} \left(\prod_{l=1}^K p_t^{j,l} \right)^{1/K}$, 选 $h_t(x)$.
 - b) 调整权值:

$$\omega_{i,l}^{t+1} = \frac{\omega_{i,l}^t}{Z_t} \exp(h_t(x_i, l) - \frac{1}{K} \sum_{k=1}^K h_t(x_i, k))$$
 - 3) 组合分类器为: $H(x) = \arg \max_l \{f(x, l)\}$.

1.2 多分类代价敏感 AdaBoost 算法的分析

算法 1 输出最大置信度对应标签, 其属单标签多分类学习算法. 一种想法是将组合分类器改为 $H(x) = \{k : f(x, k) \geq \frac{1}{K} \sum_{l=1}^K f(x, l)\}$, 即输出比平均置信度大的标签, 这似乎能得到多标签代价敏感学习算法, 但实际并非如此, 原因如下:

单标签分类问题的示例只有一个标签 (具有排他性), 示例 (x_i, y_i) 的错分代价可以用 $c(y_i, j)$ 来确定, 其中 $c(y_i, j) \geq 0$, $c(y_i, y_i) = 0$. 而多标签分类问题的样本为 (x_i, Y_i) , $Y_i \subseteq L = \{1, 2, \dots, K\}$ 为 x_i 的标签集, 显然已无法再使用代价矩阵 $c(i, j)_{K \times K}$ 来表示错分代价了.

如前所述, 预测标签集与真实标签集不相等就算错误将引出 2^K 个代价, 不具有实用性. 如果对应每个样本定义一个错分代价, 样本 (x_i, Y_i) 的错分代价定义为 $c(i, j) \geq 0$ (当 $j \notin Y_i$), 否则 $c(i, j) = 0$, $i = 1, 2, \dots, K$, 则与式 (1) 类似的平均错分代价为

$$\varepsilon_{cs} = \sum_{i=1}^m \left(\omega_i \sum_{l=1}^K c(i, l) \mathbb{I}[l \in H(x_i)] \right) \quad (3)$$

基于最小化式 (3) 来学习将得到 $H(x) = \emptyset$, 这表明最小化式 (3) 来构造的学习算法将会失效.

多标签分类问题通常被转换成多个单标签分类问题来处理, 类似地, 将多标签代价敏感分类问题转换为多个单标签代价敏感分类问题来处理, 但这同样会引出新问题. 如幂集法^[18] 将引出 2^K 个类别的多分类问题并且无法合并错分代价, 而一对多分解法^[19] 不仅导致确定错分代价困难, 还会完全忽略标签的相关性. 事实上, 考虑标签之间的相关性是成功构造多标签代价敏感分类算法的关键. 上述分析表明, 简单地基于多分类代价敏感学习算法来扩展多标签代价敏感分类算法是困难的.

2 多标签代价敏感分类集成学习算法

2.1 多标签代价敏感分类集成学习的构造

设样本集 $S = \{(x_1, Y_1), (x_2, Y_2), \dots, (x_m, Y_m)\}$, $Y_i \subseteq L$ 为示例 x_i 的标签集, $L = \{1, 2, \dots, K\}$. 弱分类器 $h_t(x)$ 对应标签 l 的置信度为 $h_t(x_i, l)$, 输出标签集 $\{h_t(x)\} = \{l : h_t(x, l) - v_t(x) \geq 0\}$, $v_t(x)$ 为阈值, 文献 [12–14] 将平均置信度作为阈值, 即 $v_t(x) = \bar{h}_t(x) = \frac{1}{K} \sum_{l=1}^K h_t(x, l)$. 实际上可以直接把 $h_t(x) - v_t(x)$ 作为弱分类器并仍记为 $h_t(x)$, 此时输出标签集为 $\{h_t(x)\} = \{l : h_t(x, l) \geq 0\}$, 而组合分类器 $H(x) = \{f(x)\} = \{l : f(x, l) \geq 0\}$, 其中 $f(x, l) = \sum_{t=1}^T h_t(x, l)$.

在多标签分类问题中, 示例的标签集或者包含标签 l 或者不包含标签 l , 即针对具体的每个标签, 多标签分类问题就变成二分类问题, 多标签分类错误包含如下 2 种: 1) 真实标签集包含的标签不在预测标签集中; 2) 真实标签集不包含的标签反而在预测标签集中, 这在单标签分类问题中是不可能出现的. 多标签分类的错分代价只需要同时考虑这 2 种情况即可.

设 $c_0(l)$ 和 $c_1(l)$ 分别为 $H(x)$ 误检 (多输出) 标

签 l 和漏检 (少输出) 标签 l 的代价, 则平均代价为

$$\varepsilon_{cs} = \sum_{i=1}^m \sum_{l=1}^K \left(\omega_i c_0(l) \llbracket l \in H(x_i) \rrbracket \llbracket l \notin Y_i \rrbracket \right) + \sum_{i=1}^m \sum_{l=1}^K \left(\omega_i c_1(l) \llbracket l \notin H(x_i) \rrbracket \llbracket l \in Y_i \rrbracket \right) \quad (4)$$

其中, $\omega_i = 1/(mK)$. 记 $b_0(i, l) = \llbracket l \notin Y_i \rrbracket$, $b_1(i, l) = 1 - b_0(i, l)$, 于是有

$$\begin{aligned} \varepsilon_{cs} &= \sum_{i=1}^m \sum_{l=1}^K \left(\omega_i c_0(l) b_0(i, l) \llbracket f(x_i, l) \geq 0 \rrbracket \right) + \\ &\sum_{i=1}^m \sum_{l=1}^K \left(\omega_i c_1(l) b_1(i, l) \llbracket f(x_i, l) < 0 \rrbracket \right) \leq \\ &\sum_{i=1}^m \sum_{l=1}^K \left(\omega_i c_0(l) b_0(i, l) \exp(f(x_i, l)) \right) + \\ &\sum_{i=1}^m \sum_{l=1}^K \left(\omega_i c_1(l) b_1(i, l) \exp(-f(x_i, l)) \right) = \\ &Z_0 \sum_{i=1}^m \sum_{l=1}^K \left(\omega_{i,l}^{1,1} \prod_{t=1}^T \exp(h_t(x_i, l)) \right) + \\ &Z_0 \sum_{i=1}^m \sum_{l=1}^K \left(\omega_{i,l}^{1,2} \prod_{t=1}^T \exp(-h_t(x_i, l)) \right) \quad (5) \end{aligned}$$

其中, $\omega_{i,l}^{1,1} = (\omega_i c_0(l) b_0(i, l))/Z_0$, $\omega_{i,l}^{1,2} = (\omega_i c_1(l) \times b_1(i, l))/Z_0$, Z_0 为 $\omega_{i,l}^{1,1} + \omega_{i,l}^{1,2}$ 的归一化因子, $Z_0 = \sum_{i=1}^m \sum_{l=1}^K (\omega_i c_0(l) b_0(i, l) + \omega_i c_1(l) b_1(i, l))$.

$$\begin{aligned} Z_1 &= \sum_{i=1}^m \sum_{l=1}^K \left(\omega_{i,l}^{1,1} \exp(h_1(x_i, l)) \right) + \\ &\sum_{i=1}^m \sum_{l=1}^K \left(\omega_{i,l}^{1,2} \exp(-h_1(x_i, l)) \right) \quad (6) \end{aligned}$$

在式 (5) 中提取出 Z_1 , 则式 (5) 变为

$$\begin{aligned} Z_0 Z_1 &\sum_{i=1}^m \sum_{l=1}^K \left(\omega_{i,l}^{2,1} \prod_{t=2}^T \exp(h_t(x_i, l)) \right) + \\ &Z_0 Z_1 \sum_{i=1}^m \sum_{l=1}^K \left(\omega_{i,l}^{2,2} \prod_{t=2}^T \exp(-h_t(x_i, l)) \right) \quad (7) \end{aligned}$$

其中, $\omega_{i,l}^{2,1} = \omega_{i,l}^{1,1} \exp(h_1(x_i, l))/Z_1$, $\omega_{i,l}^{2,2} = \omega_{i,l}^{1,2} \times \exp(-h_1(x_i, l))/Z_1$. Z_1 是 $\omega_{i,l}^{2,1} + \omega_{i,l}^{2,2}$ 的归一化因子.

为使式 (7) 取到极小值, 可基于最小化 Z_1 学习 $h_1(x)$, 而式 (7) 与式 (5) 形式相同, 于是学习得到

$h_1(x)$ 后, 可以类似地学习 $h_2(x)$, 由此便可递推式学习后续各个弱分类器.

分类器的本质是对目标空间的一个划分, 在相同划分段内的示例有相同的标签. 设 $h_1(x)$ 对应的划分为 $S = S_1^1 \cup S_2^1 \cup \dots \cup S_{n_1}^1$, $i \neq j$ 时, $S_i^1 \cap S_j^1 = \emptyset$. 当 $x_i \in S_j^1$ 时, $h_1(x_i)$ 输出标签 l 的置信度 $h_1(x_i, l)$ 记为 $\alpha_1^{j,l}$, $j = 1, 2, \dots, n_1$. 合并 Z_1 中同划分段内相同项, 记 $p_1^{j,l} = \sum_{i:(x_i \in S_j^1)} \omega_{i,l}^{1,1}$, $q_1^{j,l} = \sum_{i:(x_i \in S_j^1)} \omega_{i,l}^{1,2}$, 则有

$$\begin{aligned} Z_1 &= \sum_{j=1}^{n_1} \sum_{l=1}^K (p_1^{j,l} \exp(\alpha_1^{j,l})) + \\ &\sum_{j=1}^{n_1} \sum_{l=1}^K (q_1^{j,l} \exp(-\alpha_1^{j,l})) \quad (8) \end{aligned}$$

求取 $p_1^{j,l} \exp(\alpha_1^{j,l}) + q_1^{j,l} \exp(-\alpha_1^{j,l})$ 的极小值, 可得其极小值点为 $\alpha_1^{j,l} = 0.5 \ln(q_1^{j,l}/p_1^{j,l})$, 此时 $Z_1 = 2 \sum_{j=1}^{n_1} \sum_{l=1}^K \sqrt{p_1^{j,l} q_1^{j,l}}$, 于是有如下的多标签代价敏感分类集成学习算法 (Cost-sensitive AdaBoost for multi-label problems, Cost-MLPBoost).

算法 2. 多标签代价敏感分类集成学习算法

1) 初始化权值: $\omega_{i,l}^{1,1} = \omega_i c_0(l) b_0(i, l)/(Z_0)$, $\omega_{i,l}^{1,2} = \omega_i c_1(l) b_1(i, l)/(Z_0)$, Z_0 是 $\omega_{i,l}^{1,1} + \omega_{i,l}^{1,2}$ 的归一化因子, $i = 1, 2, \dots, m$, $l = 1, 2, \dots, K$;

2) Do for $t = 1, 2, \dots, T$;

a) 训练弱分类器:

i) 计算 $p_t^{j,l}$, $q_t^{j,l}$: 对应划分 $S = S_1^t \cup \dots \cup S_{n_t}^t$, 计算 $p_t^{j,l} = \sum_{i:(x_i \in S_j^t)} \omega_{i,l}^{t,1}$, $q_t^{j,l} = \sum_{i:(x_i \in S_j^t)} \omega_{i,l}^{t,2}$, $j = 1, 2, \dots, n_t$;

ii) 定义 $h_t(x)$: $h_t(x, l) = 0.5 \ln(q_t^{j,l}/p_t^{j,l})$, $\forall x \in S_j^t$;

iii) 选取 $h_t(x)$:

最小化 $Z_t = 2 \sum_{j=1}^{n_t} \sum_{l=1}^K \sqrt{p_t^{j,l} q_t^{j,l}}$ 选 $h_t(x)$.

b) 调整权值:

$$\omega_{i,l}^{t+1,1} = \frac{\omega_{i,l}^{t,1}}{Z_t} \exp(h_t(x_i, l)),$$

$$\omega_{i,l}^{t+1,2} = \frac{\omega_{i,l}^{t,2}}{Z_t} \exp(-h_t(x_i, l));$$

3) 组合分类器为:

$$H(x) = \{f(x)\} = \{l : f(x, l) \geq 0\}.$$

由算法的推导过程可以得到算法 2 的平均错分代价上界估计为

$$\varepsilon_{cs} \leq \prod_{t=0}^T Z_t = Z_0 \prod_{t=1}^T \left(2 \sum_{j=1}^{n_t} \sum_{l=1}^K \sqrt{p_t^{j,l} q_t^{j,l}} \right) \quad (9)$$

$$\text{根据不等式 } Z_t \leq \sum_{j=1}^{n_t} \sum_{l=1}^K (p_t^{j,l} + q_t^{j,l}) =$$

$\sum_{i=1}^m \sum_{l=1}^K (\omega_{i,l}^{t,1} + \omega_{i,l}^{t,2}) = 1$, 当且仅当 $p_t^{j,l} = q_t^{j,l}$ 对所有 $j \in \{1, 2, \dots, n_t\}$ 和 $l \in \{1, 2, \dots, K\}$ 都成立时才会有 $Z_t = 1$, 算法 2 满足“ $Z_t \leq 1$ 且一般情况下 $Z_t < 1$ ”条件, 因此, 算法 2 的平均错分代价能随着弱分类器的增加而逐渐降低.

算法 2 的归一化因子 Z_t 由全部训练样本的所有标签对应的权值之和构成, 因此, 当标签之间有一定的相关性时, 其将在该归一化因子中得以体现, 所以, 算法 2 在一定程度上考虑了标签之间的相关性.

进一步分析发现, 在每轮学习中, 算法 2 都将样本的两组权值和调为相等, 即 $\sum_{i=1}^m \sum_{l=1}^K \omega_{i,l}^{t,1} = \sum_{i=1}^m \sum_{l=1}^K \omega_{i,l}^{t,2}$. 如果令 $h_t(x, l) = 0.25 \ln(q_t^{j,l}/p_t^{j,l})$, 则可以破坏两组权值和相等现象, 于是肯定不会出现 $Z_t = 1$, 理论上这将确保每新增加一个弱分类器都能进一步降低错分代价. 于是, 可以修改算法 2 中的弱分类器定义为

$$h_t(x, l) = 0.25 \ln(q_t^{j,l}/p_t^{j,l}) \quad (10)$$

此时有 $Z_t = \sum_{j=1}^{n_t} \sum_{l=1}^K (p_t^{j,l})^{\frac{1}{4}} (q_t^{j,l})^{\frac{3}{4}} + \sum_{j=1}^{n_t} \sum_{l=1}^K (p_t^{j,l})^{\frac{3}{4}} (q_t^{j,l})^{\frac{1}{4}}$. 因为 $(p_t^{j,l})^{\frac{1}{4}} (q_t^{j,l})^{\frac{3}{4}} + (p_t^{j,l})^{\frac{3}{4}} (q_t^{j,l})^{\frac{1}{4}} \leq p_t^{j,l} + q_t^{j,l}$, 所以此时的算法仍然满足“ $Z_t \leq 1$ 且一般情况下 $Z_t < 1$ ”条件, 即平均错分代价仍然能随着弱分类器的增加而逐渐降低.

2.2 算法 2 与算法 1 的比较分析

2.2.1 算法 2 与算法 1 的区别

1) 输出标签 (类别) 依据的区别

算法 1 输出最大置信度对应的标签, 根据最大置信度一定比平均置信度大这一特点, 并利用不等式 $\mathbb{I}[H(x_i) = l] \leq \exp(f(x_i, l) - \bar{f}(x_i))$, 把用符号函数表示的平均错分代价转换成用指数函数来表示, 进而将分类器之和转换成分类器的指数函数之积, 最终构造出了递推算法. 错分代价相等时, 算法 1 的本质是: 统计划分段内不包含每类示例的概率, 然后输出概率最小的类别.

出乎预想的是, 算法 2 输出的标签并不是比平均置信度大的标签, 实际上, 算法 2 的第二组权值 $\omega_{i,l}^{t,2}$ 代表了包含标签的比率, 第一组权值 $\omega_{i,l}^{t,1}$ 代表了不包含标签的比率, $h_t(x, l) > 0$ 输出标签 l 意味着 $q_t^{j,l} > p_t^{j,l}$ 输出标签 l . 可见, 算法 2 是比较包含标签比率和不包含标签的比率来决定输出, 前者大则输出该标签, 反之不输出该标签. 错分代价相等时, 算法 2 的本质是: 统计划分段包含标签 l 的概率和不包含标签 l 的概率, 前者大则输出标签 l , 反之不输出标签 l . 显然, 算法 1 和算法 2 具有本质的不同.

2) 错分代价的区别

多分类代价敏感问题的错分代价可以根据示例的类别来确定, i 类被分为 j 类的代价 $c(i, j)$ 能准确度量各种错分情况, 一共有 $K^2 - K$ 个.

多标签代价敏感分类问题的错分代价分为误检标签和漏检标签两类, 用 $c_0(l)$ 和 $c_1(l)$ 来度量误检标签 l 和漏检标签 l 的代价, 共有 $2K$ 个, 可见两个算法的错分代价是不同的. 实际上, 两个算法是从不同的角度来控制学习结果, 算法 1 通过错分为其他类的相对代价来控制学习, 其结果是尽量把示例分为错分代价相对较小的类, 而算法 2 通过单个标签被错分的代价来控制学习; 其结果是尽量向误检代价和漏检代价相对较小的标签倾斜.

2.2.2 用于单标签分类的算法 2 与算法 1 的比较

算法 2 可用于解决一般单标签代价敏感分类, 只需改算法 2 的 $H(x) = \arg \max_l \{f(x, l)\}$ 即可. 此时, $c_0(l)$ 和 $c_1(l)$ 分别表示非 l 类示例被错分为 l 类的代价和 l 类示例被错分为其他类的代价, 其与算法 1 的错分代价可以相互转换, $c_0(l)$ 可以取被错分为 l 类的代价 ($K-1$ 个) 的平均值, $c_1(l)$ 可以取 l 类被错误分为其他类 ($K-1$ 个) 的代价的平均值, 即

$$c_0(l) = \sum_{k=1}^K \left(c(k, l) / (K-1) \right) \quad (11)$$

$$c_1(l) = \sum_{k=1}^K \left(c(l, k) / (K-1) \right) \quad (12)$$

反过来的转换公式为

$$c(i, j) = c_1(i) + c_0(j) \quad (13)$$

尽管用于单标签分类的算法 2 与算法 1 的代价可以相互转换, 但二者仍有很大的不同. 算法 1 的代价矩阵 $c(i, j)_{K \times K}$ 能区分每一类被错分为其他类的代价, 学习结果将是: $\min_j \{c(i, j)\}, i = 1, 2, \dots, K$ 较大的类尽量不要被错分, 即使被错分也尽量错分到 $c(i, j), j = 1, 2, \dots, K$ 较小的类. 用于单标签分类的算法 2 中 $c_0(l)$ 表示被错分为 l 类的代价 (真实类为 $k, k \neq l$), 而 $c_1(l)$ 表示 l 类被错分为其他类的代价, 学习结果将是: $c_1(l)$ 较大的类尽量不要被错分, 即使被错分也将尽量错分到 $c_0(l)$ 较小的类.

当错分代价相等时, 算法 2 将简化为 Hamming loss 最小化的多标签分类集成学习算法, 该算法可以确保组合分类器的 Hamming loss 随着弱分类器的增加而逐渐降低.

2.3 错分代价限制条件分析

多分类代价敏感学习算法一般要求错分代价大

于 0 即可. 多标签代价敏感学习算法对错分代价的要求复杂得多, $c_0(l) \geq 0$ 和 $c_1(l) \geq 0$ 仅仅是基本要求. 由于两个畸形分类器 $H(x) = L$ 和 $H(x) = \emptyset$ 是合理分类器, 如果对错分代价不加以限制, 学习到的分类器极有可能是这两个畸形分类器之一. 比如, 只考虑误检标签的损失, $H(x) = \emptyset$ 就是错分代价最小的分类器; 而只考虑漏检标签的损失, $H(x) = L$ 就是错分代价最小的分类器. 单标签分类问题不会出现这种情况. 因此, 详细分析多标签代价敏感学习算法的错分代价非常重要.

先考虑不同标签错分代价相等情况, 设 $c_0(l) = c_0$, $c_1(l) = c_1$, d 为标签密度.

首先, 如果 $c_0 K(1-d) < c_1$, 算法将失效. 原因如下: 当 $H(x) = L$ 时, 分类器平均误检标签个数为 $K(1-d)$, 其代价为 $c_0 K(1-d)$, $c_0 K(1-d) < c_1$ 表示分类器输出全部标签的平均代价反而比漏检 1 个标签的代价小, 算法的学习结果势必会趋于 $H(x) = L$, 这将导致算法失效.

其次, 如果 $c_0 > Kdc_1$, 算法也将失效. 原因是当 $H(x) = \emptyset$ 时, 分类器平均漏检标签个数为 Kd , 其代价为 $c_1 Kd$, $c_0 > Kdc_1$ 表示分类器不输出任何标签的平均代价反而比误检 1 个标签的代价小, 算法的学习结果势必会趋于 $H(x) = \emptyset$, 这也将导致算法失效.

代价同乘以一个常数不影响算法学习结果, 不失一般性, 假设 $c_1 + c_0 = 1$. 根据前面的分析可知, 多标签代价敏感分类问题的错分代价限制条件为

$$1/(K(1-d) + 1) < c_0 < Kd/(1 + Kd) \quad (14)$$

可类似地分析不同标签错分代价不等情况. 设 P_l 为标签 l 的密度 (标签集包含标签 l 的比率), $H(x) = L$ 时的平均代价为 $\sum_{l=1}^K (1-P_l)c_0(l)$, 而漏检 1 个标签的平均代价为 $\frac{1}{K} \sum_{l=1}^K c_1(l)$, 当 $\sum_{l=1}^K (1-P_l)c_0(l) < \frac{1}{K} \sum_{l=1}^K c_1(l)$ 时, 算法的学习结果将趋于 $H(x) = L$. 同理, 当 $\frac{1}{K} \sum_{l=1}^K c_0(l) > \sum_{l=1}^K P_l c_1(l)$ 时, 算法的学习结果将趋于 $H(x) = \emptyset$. 于是一般情形的多标签代价敏感分类问题的错分代价限制条件为

$$\sum_{l=1}^K ((1-P_l)c_0(l)) > (1/K) \sum_{l=1}^K c_1(l) \quad (15)$$

$$(1/K) \sum_{l=1}^K c_0(l) < \sum_{l=1}^K (P_l c_1(l)) \quad (16)$$

上述表达式过于复杂, 简单的近似限制条件为

$$c_0(l)(1-P_l) > c_1(l)/K \quad (17)$$

$$c_0(l)/K < P_l c_1(l) \quad (18)$$

需要注意的是, 此时没有 $c_0(l) + c_1(l) = 1$ 条件.

3 实验与分析

3.1 实验方法和数据

实验主要验证 Cost-MLPBoost 的有效性、错分代价限制条件的合理性和用于解决单标签代价敏感问题时的效果. 实验数据来源于 Mulan¹ 和 UCI^[26], 具体见表 1.

表 1 实验数据集

Table 1 The experimental data

数据集	示例总数	属性数	标签数	标签基数	标签密度
Emotions	593	72	6	1.869	0.311
Scene	2 407	294	6	1.074	0.179
Yeast	2 417	103	14	4.237	0.303
Wine	178	14	3	1	0.333
Ionosphere	351	34	2	1	0.5

弱分类器基于单个属性来构造, 即基于样本的轴向投影来构造, 弱分类器采取 4 段划分, 3 个分段阈值为: 含有相同标签的样本均值, 最大值与该均值的平均, 最小值与该均值的平均. 为有更多的弱分类器可供选择, 用加权均值代替上述均值, x_i 的加权系数为 $u_i^t = (1/\sum_{l=1}^K \omega_{i,l}^{t,1} + \sum_{l=1}^K \omega_{i,l}^{t,2})/U_t$, U_t 为 u_i^t 的归一化因子. 理由如下: 错分样本的第一个权值将变小而第二个权值将变大, 上述加权系数选取策略可以使分类器向被错分的样本聚焦. 实验按照 7:3 比例随机划分数据得训练集和测试集, 弱分类器个数 $T = 20$, 重复 10 次, 统计平均代价或错误率, 实验时采用算法 2 (弱分类器由式 (10) 给出).

3.2 实验结果与分析

1) Cost-MLPBoost 的有效性验证和错分代价限制条件验证.

考虑简单情况, $c_0(l) = c_0$, $c_1(l) = c_1$, $c_0 + c_1 = 1$. 实验时, 先计算出各实验数据集的有效代价取值区和无效代价取值区, 在两个区域均匀各取 4 个代价进行实验. 实验结果见图 1~6 所示.

据式 (14) 计算可知, Emotions 数据集的 c_0 的有效取值区域为 $[0.195, 0.65]$, 该区域内的 4 个错分代价对应的实验结果见图 1. 结果表明, 组合分类器的平均错分代价随着弱分类器的增加而逐渐降低, $T = 20$ 时比 $T = 1$ 时的平均错分代价下降了约 18%, 表明算法是有效的. c_0 的无效区域 $[0, 0.195]$ 和 $[0.65, 1]$ 内 4 个错分代价对应的实验结果见图 2, 结果表明, 算法已经失效, 随着弱分类器的增加, 组

¹<http://mulan.sourceforge.net/datasets.html>

合分类器的平均错分代价并没有明显降低.

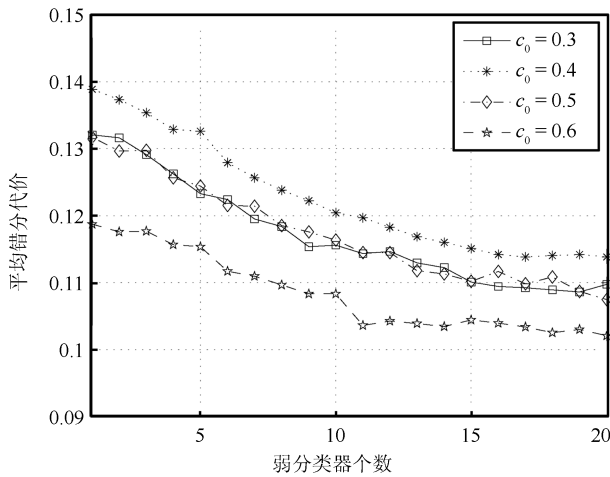


图1 基于有效的 c_0 在 Emotions 数据上的平均代价

Fig. 1 Average of cost in Emotions on valid c_0

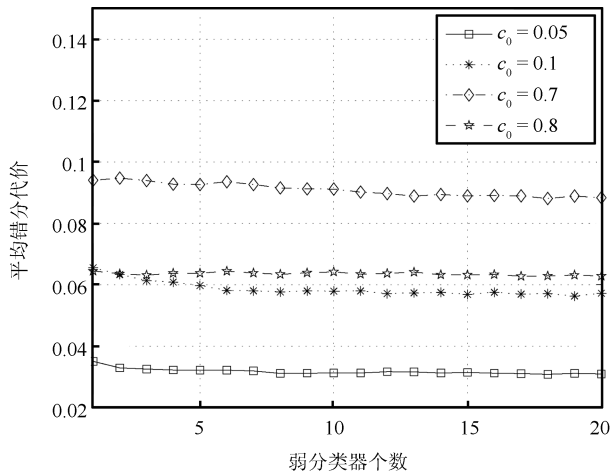


图2 基于无效的 c_0 在 Emotions 数据上的平均代价

Fig. 2 Average of cost in Emotions on invalid c_0

关于错分代价的无效性还可以进行如下分析:

分析 $c_0 = 0.05$ 情况. 当 $H(x) = L$ 时将平均多输出 4.13 个标签, 其平均错分代价为 $4.13 \times 0.05/6 = 0.0344$, 图 2 显示此时组合分类器的平均错分代价近似于该值, 表明学习所得分类器近似等于 $H(x) = L$, 算法失效. 分析 $c_0 = 0.7$ 情况, 此时, $c_1 = 0.3$, $H(x) = \emptyset$ 时的平均错分代价为 $1.87 \times 0.3/6 = 0.093$, 图 2 中 $c_0 = 0.7$ 的实验结果 ($T = 1$) 取值 0.094, 表明学习所得分类器近似等于 $H(x) = \emptyset$, 算法失效. 同理, $c_0 = 0.8$ 时将得到 $H(x) = \emptyset$; $c_0 = 0.1$ 时, 将得到 $H(x) = L$, 算法都将失效.

Scene 数据的 c_0 有效取值区域为 $[0.168, 0.52]$, 该区域内的 4 个错分代价对应实验结果见图 3. 结果表明, 组合分类器的平均错分代价随着弱分类器的增加而逐渐降低, $T = 20$ 比 $T = 1$ 的平均错分代

价下降了约 40%, 算法有效. 图 4 则显示了 4 个取值于无效错分代价区域的实验结果, 结果表明算法失效.

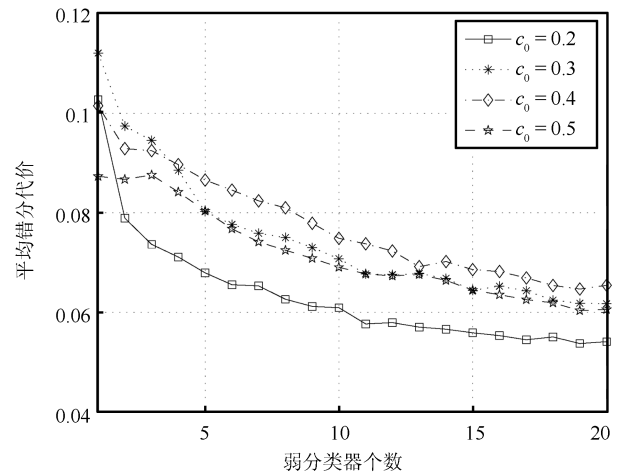


图3 基于有效的 c_0 在 Scene 数据上的平均代价

Fig. 3 Average of cost in Scene on valid c_0

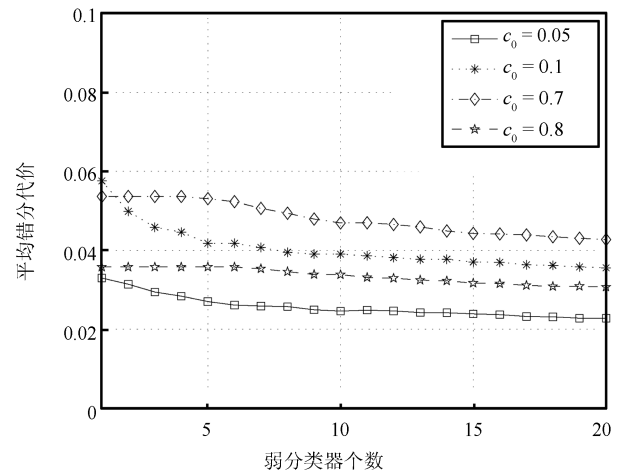


图4 基于无效的 c_0 在 Scene 数据上的平均代价

Fig. 4 Average of cost in Scene on invalid c_0

Yeast 数据的 c_0 有效取值区域为 $[0.093, 0.8]$, 图 5 和图 6 分别显示 4 个有效错分代价和 4 个无效错分代价对应的实验结果, 结果同样表明了错分代价取值于有效区域时算法有效, 否则算法将失效.

2) Cost-MLPBoost 利用标签相关性的验证和与基于二分类代价敏感的多标签 AdaBoost 算法 (AdaBoost-based cost-sensitive binary tag classifier, AdaBoost-CSML)^[27] 的比较.

Cost-MLPBoost 与组合式多标签 AdaBoost 算法 (Multi-label AdaBoost based on combining multiple binary AdaBoost, Com-AdaBoost) (基于 K 个独立的二分类 AdaBoost 算法采取合并输出结果) 的比较实验结果见表 2 ($c_0 = 0.5$), 后者完全没

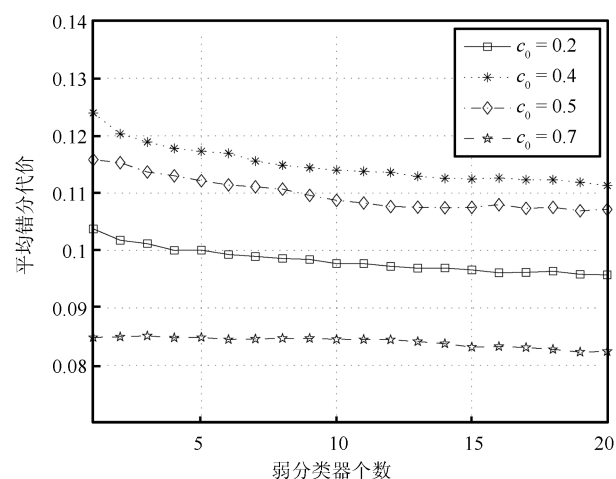


图 5 基于有效的 c_0 在 Yeast 数据上的平均代价
Fig. 5 Average of cost in Yeast on valid c_0

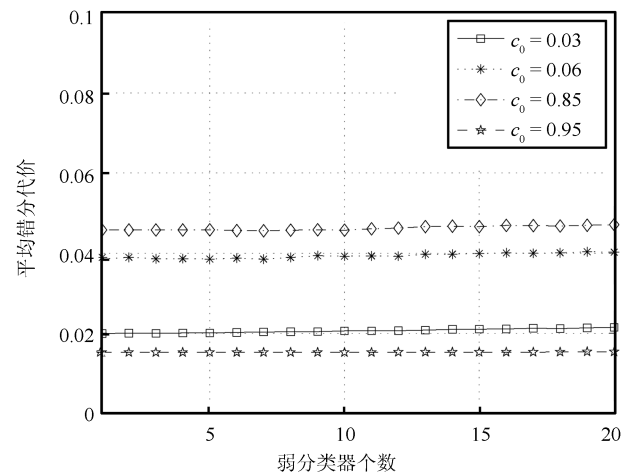


图 6 基于无效的 c_0 在 Yeast 数据上的平均代价
Fig. 6 Average of cost in Yeast on invalid c_0

表 2 Cost-MLPBoost 与 Com-AdaBoost 的实验结果比较
Table 2 Comparison of Cost-MLPBoost with Com-AdaBoost

数据集	算法	$T = 1$	$T = 10$	$T = 20$	$T = 30$	$T = 40$
Emotions	Cost-MLPBoost	0.1477	0.1134	0.1078	0.1055	0.1054
	Com-AdaBoost	0.1461	0.1300	0.1261	0.1245	0.1241
Scene	Cost-MLPBoost	0.0882	0.0732	0.0584	0.0542	0.0511
	Com-AdaBoost	0.0892	0.0838	0.0801	0.0746	0.0709

表 3 不同代价下的 Cost-MLPBoost 与 AdaBoost-CSML 比较
Table 3 Comparison of Cost-MLPBoost with AdaBoost-CSML on different cost

数据集	$c_0(1)$	$c_0(2)$	$c_0(3)$	$c_0(4)$	$c_0(5)$	$c_0(6)$	Cost-MPLBoost	AdaBoost-CSML
Emotions	0.2	0.3	0.4	0.5	0.6	0.7	0.1023	0.1140
	0.7	0.6	0.5	0.4	0.3	0.2	0.0987	0.1119
Scene	0.2	0.25	0.3	0.35	0.4	0.45	0.0537	0.0744
	0.45	0.4	0.35	0.3	0.25	0.2	0.0535	0.0709

表 4 在 Emotions 数据集上基于不同代价的实验结果
Table 4 The experimental results in Emotions data on different cost

实验	l	1	2	3	4	5	6	l	1	2	3	4	5	6
1	$c_0(l)$	1	2	3	4	5	6	$over(l)$	0.0702	0.0028	0.0506	0.0096	0.0022	0.0006
	$c_1(l)$	1	1	1	1	1	1	$less(l)$	0.1517	0.2674	0.3230	0.1539	0.2708	0.3124
2	$c_0(l)$	1	2	3	4	5	6	$over(l)$	0.2921	0.2556	0.1522	0.0337	0.0084	0.0006
	$c_1(l)$	6	5	4	3	2	1	$less(l)$	0.0393	0.1191	0.1213	0.0955	0.2298	0.3140
3	$c_0(l)$	1	2	3	4	5	6	$over(l)$	0.3079	0.3197	0.1916	0.0635	0.0556	0.0904
	$c_1(l)$	6	6	6	6	6	6	$less(l)$	0.0360	0.1006	0.0815	0.0567	0.1354	0.1090
4	$c_0(l)$	1	1	1	1	1	1	$over(l)$	0.0652	0.1461	0.2320	0.1315	0.3219	0.2539
	$c_1(l)$	1	2	3	4	5	6	$less(l)$	0.1393	0.1579	0.0500	0.0287	0.0528	0.0208
5	$c_0(l)$	6	5	4	3	2	1	$over(l)$	0	0.0017	0.1101	0.0629	0.2011	0.2551
	$c_1(l)$	1	2	3	4	5	6	$less(l)$	0.2876	0.2865	0.1888	0.0680	0.0921	0.0157

有利用标签的相关性. 结果表明, Cost-MLPBoost 好于 Com-AdaBoost, 这在一定程度上验证了本文算法考虑了标签的相关性.

而 Cost-MLPBoost 与文献 [27] 提到的 AdaBoost-CSML 的比较实验结果见表 3. 实验中错分代价均取值于有效代价区域, Emotions 为 $[0.195, 0.65]$, Scene 为 $[0.168, 0.52]$.

表 3 所示对比实验结果表明 Cost-MLPBoost 好于 AdaBoost-CSML.

3) Cost-MLPBoost 随着错分代价的变化而变化的实验分析.

代价敏感分类的一个重要应用就是通过人为控制错分代价以使得分类器向期望类倾斜或输出期望的标签, 这也是代价敏感分类算法所必须具备的功能. $c_0(l)$ 和 $c_1(l)$ 不同取值的实验结果见表 4, 其中 $over(l)$ 为误检标签 l 的比率, $less(l)$ 为漏检标签 l 的比率.

表 4 中实验 1 的 $c_0(l) = l$, $c_1(l) = 1$, 即漏检任何标签的代价一样, 而误检不同标签的代价不同, 其中误检标签 6 的代价最大, 于是理想算法应该向误检代价较小的标签集中, 即 $over(6)$ 应该最小而 $over(1)$ 应该最大. 实验结果与期望完全吻合, 且 $over(l)$ 随 l 增加而逐渐减小; 对应的 $less(6)$ 最大而 $less(1)$ 最小.

实验 2 的 $c_0(l) = l$, $c_1(l) = 7 - l$, 即误检标签 l 的代价随 l 增大而逐渐增大, 相反漏检标签 l 的代价逐渐减小, 于是对如标签 6 这种误检代价大而漏检代价小的标签, 一定是期望“宁少勿多”. 实验结果表现出了这一现象, 其中 $over(l)$ 随 l 增加而逐渐减小 (0.2921 到 0.0006), 而且标签 6 几乎不误检.

实验 3 的 $c_0(l) = l$, $c_1(l) = 6$, 即漏检任何标签的代价都相对较大, 于是学习结果将是“宁多勿少”, 实验结果很好地表现出了这一现象.

实验 4 的 $c_0(l) = 1$, $c_1(l) = l$, 与实验 1 相反, 误检任何标签的代价一样, 而漏检不同标签的代价不同, 实验结果表现为 $less(l)$ 随 l 增加而逐渐减小.

实验 5 的 $c_0(l) = 7 - l$, $c_1(l) = l$, 期望是“宁少勿多”, 实验结果与期望吻合. 其中 $over(1) = 0$ 表示算法根本就不会误检标签 1, 值得注意的是, 误检率太低必然会导致产生大的漏检率, 此时其漏检率为 $less(1) = 0.2876$.

上述实验结果表明, Cost-MLPBoost 的学习结果与期望结果是一致的, 即算法能够确保平均错分代价最小化, 这进一步验证了算法 2 的有效性.

4) 算法 2 用于单标签代价敏感分类与已有单标签代价敏感学习算法的比较.

Cost-MLPBoost 可用于单标签代价敏感分类. Cost-MLPBoost、Cost-MCPBoost^[12] 和不平衡代

价调权提升算法 (Boosting with unequal initial weights on cost-sensitive adaptation)^[11] 的比较实验结果见表 5 ~ 6.

实验时, Cost-MLPBoost 与 Cost-MCPBoost 之间的代价转换按照公式 (13) 计算, 而与 Cost-UBoost (Boosting with unequal initial weights on cost-sensitive adaptation) 算法的代价转换公式采用 $c_1(l) + (1/(K-1)) \sum_{k \neq l} c_0(k)$, 即自己类标签错分代价与错分到其他类的平均代价之和.

表 5 和表 6 是在三分类数据集 Wine 上的实验结果, 主要区别为错分代价的差距不同, 表 5 的错分代价相差最多 2 倍. 实验表明, Cost-MLPBoost 好于 Cost-MCPBoost (代价降低 20%, 实验 4 例外), 两者都好于 Cost-UBoost (代价降低了 50%). 表 6 的错分代价差距较大 (约 3~7 倍).

实验结果表明, Cost-MLPBoost 好于 Cost-MCPBoost (代价降低了 20%~50%), 两者都好于 Cost-UBoost. 这说明, 当错分代价相差较大时, Cost-MLPBoost 用于解决单标签代价敏感分类问题, 能取得更好的效果.

表 7 是在二分类数据集 Ionosphere 上的实验结果. 实验结果表明, Cost-MLPBoost 好于 Cost-MCPBoost (平均代价降低约 10%), 而 Cost-MCPBoost 又好于 Cost-UBoost.

表 5 Wine 数据集上不同代价的实验 (A)

Table 5 The experiment in Wine on different cost (A)

l	1	2	3	Cost-MLPBoost	Cost-MCPBoost	Cost-UBoost
$c_0(l)$	1	1	1	0.0906	0.1132	0.1811
$c_1(l)$	1	2	3			
$c_0(l)$	3	2	1	0.1113	0.1415	0.2019
$c_1(l)$	1	2	3			
$c_0(l)$	3	3	3	0.1642	0.1906	0.2981
$c_1(l)$	1	2	3			
$c_0(l)$	1	2	3	0.1132	0.1019	0.1717
$c_1(l)$	1	1	1			

表 6 Wine 数据集上不同代价的实验 (B)

Table 6 The experiment in Wine on different cost (B)

l	1	2	3	Cost-MLPBoost	Cost-MCPBoost	Cost-UBoost
$c_0(l)$	1	4	7	0.1094	0.2415	0.3925
$c_1(l)$	1	1	1			
$c_0(l)$	1	4	7	0.1774	0.2604	0.4226
$c_1(l)$	3	2	1			
$c_0(l)$	1	1	1	0.0925	0.1623	0.1689
$c_1(l)$	1.5	2	2.5			
$c_0(l)$	1	1	1	0.0994	0.1497	0.1610
$c_1(l)$	1.3	1.6	2			

表 7 Ionosphere 数据集上不同代价的实验

Table 7 The experiment in Ionosphere on different cost

l	1	2	Cost-MLPBoost	Cost-MCPBoost	Cost-UBoost
$c_0(l)$	1	1	0.2476	0.2733	0.3133
$c_1(l)$	1	1			
$c_0(l)$	1	1	0.4486	0.4848	0.5610
$c_1(l)$	5	1			
$c_0(l)$	1	1	0.6038	0.6990	0.7905
$c_1(l)$	15	1			
$c_0(l)$	1	1	0.8210	0.9171	0.9943
$c_1(l)$	20	1			
$c_0(l)$	1	1	0.3162	0.3352	0.4629
$c_1(l)$	1	5			
$c_0(l)$	1	1	0.3886	0.4133	0.5619
$c_1(l)$	1	15			

4 结论

在分析多分类代价敏感 AdaBoost 算法基础上, 提出了一种平均错分代价最小化的多标签代价敏感分类集成学习算法. 平均错分代价采用了误检标签代价和漏检标签代价之和, 算法能确保平均错分代价随着弱分类器的增加而逐渐降低, 在一定程度上解决了多标签代价敏感分类学习问题. 分析了多标签代价敏感分类的错分代价限制条件, 并给出了基于标签密度的有效错分代价区域计算公式. 简化提出算法可得 Hamming loss 最小化的多标签分类学习算法和多分类代价敏感学习算法. 实验验证了提出算法的有效性和错分代价限制条件的正确性. 当错分代价相差较大时, 本文算法用于解决多类代价敏感分类问题时, 会取得比 Cost-MCPBoost 和 Cost-UBoost 更好的效果.

References

- Turney P. Types of cost in inductive concept learning. In: Proceedings of the Cost-Sensitive Learning Workshop at the 17th International Conference on Machine Learning. Stanford, USA: NRC, 2000. 15–21
- Ting K M. An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*, 2002, **14**(3): 659–665
- Domingos P. MetaCost: a general method for making classifiers cost-sensitive. In: Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 1999. 155–164
- Elkan C. The foundations of cost-sensitive learning. In: Proceedings of the 17th International Joint Conference of Artificial Intelligence. San Francisco, USA: Morgan Kaufmann, 2001. 973–978
- Bruka I, Kocková S. A support for decision-making: cost-sensitive learning system. *Artificial Intelligence in Medicine*, 1999, **6**(7): 67–82
- Zadrozny B, Langford J, Abe N. Cost-sensitive learning by cost-proportionate example weighting. In: Proceedings of the 3rd IEEE International Conference on Data Mining. Washington D. C., USA: IEEE, 2003. 435–442
- Ling C X, Sheng V S, Yang Q. Test strategies for cost-sensitive decision trees. *IEEE Transactions of Knowledge and Data Engineering*, 2006, **18**(8): 1055–1067
- Chai X Y, Deng L, Yang Q, Ling C X. Test-cost sensitive Naive Bayes classification. In: Proceedings of the 4th IEEE International Conference on Data Mining. Washington, D. C., USA: IEEE, 2004. 51–58
- Ling C X, Sheng V S. A comparative study of cost-sensitive classifiers. *Chinese Journal of Computers*, 2007, **30**(8): 1203–1211
- Ting K M, Zheng Z. Boosting cost-sensitive trees. In: Proceedings of the 1st International Conference on Discovery Science. London, UK: Springer, 1999. 244–255
- Fan W, Stolfo S J, Zhang J, Chan P K. AdaCost: misclassification cost-sensitive boosting. In: Proceedings of the 16th International Conference on Machine Learning. San Francisco, CA: Morgan Kaufmann, 1999. 97–105
- Fu Zhong-Liang. Cost-sensitive AdaBoost algorithm for multi-class classification problems. *Acta Automatica Sinica*, 2011, **37**(8): 973–983
(付忠良. 多分类问题代价敏感 AdaBoost 算法. 自动化学报, 2011, **37**(8): 973–983)
- Fu Zhong-Liang. An ensemble learning algorithm for direction prediction. *Shanghai Jiaotong University (Science Edition)*, 2012, **46**(2): 250–258
(付忠良. 一种用于方向预测的集成学习算法. 上海交通大学学报 (自然版), 2012, **46**(2): 250–258)
- Fu Zhong-Liang. A universal ensemble learning algorithm. *Journal of Computer Research and Development*, 2013, **50**(4): 861–872
(付忠良. 通用集成学习算法的构造. 计算机研究与发展, 2013, **50**(4): 861–872)
- Tsoumakas G, Katakis I. Multi-label classification: an overview. *International Journal of Data Warehousing and Mining*, 2007, **3**(3): 1–13
- Zhou Z H, Zhang M L, Huang S J, Li Y F. Multi-instance multi-label learning. *Artificial Intelligence*, 2012, **176**(1): 2291–2320

- 17 Zhang M L, Zhou Z H. M3MIML: a maximum margin method for multi-instance multi-label learning. In: Proceedings of the 8th IEEE International Conference on Data Mining. Pisa, Italy: IEEE, 2008. 688–697
- 18 Trohidis K, Tsoumakas G, Kalliris G, Vlahavas I. Multi-label classification of music into emotions. In: Proceedings the 9th International Conference on Music Information Retrieval. Philadelphia, USA: Springer, 2008. 325–330
- 19 Boutell M R, Luo J B, Shen X P, Brown C M. Learning multi-label scene classification. *Pattern Recognition*, 2004, **37**(9): 1757–1771
- 20 Zhang M L, Zhou Z H. ML-KNN: a lazy learning approach to multi-label learning. *Pattern Recognition*, 2007, **40**(7): 2038–2048
- 21 Elisseeff A, Weston J. A kernel method for multi-labeled classification. In: Proceedings of Advances in Neural Information. Cambridge: MIT, 2001, 681–687
- 22 Yin Hui, Xu Jian-Hua, Xu Hua. A multi-label classification algorithm based on LS-SVM. *Journal of Nanjing Normal University (Engineer and Technology Edition)*, 2010, **10**(2): 68–73
(殷会, 许建华, 许花. 基于 LS-SVM 的多标签分类算法. 南京师范大学学报 (工程技术版), 2010, **10**(2): 68–73)
- 23 Benhouzid D, Busa-Fekete R, Cadagrande N, Collin F D, Kégl B. MultiBoost: a multi-purpose boosting package. *Journal of Machine Learning Research*, 2012, **13**: 549–553
- 24 Cao Ying, Miao Qi-Guang, Liu Jia-Chen, Gao Lin. Advance and prospects of AdaBoost algorithm. *Acta Automatica Sinica*, 2013, **39**(6): 745–758
(曹莹, 苗启广, 刘家辰, 高琳. AdaBoost 算法研究进展与展望. 自动化学报, 2013, **39**(6): 745–758)
- 25 Schapire R E, Singer Y. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 1999, **37**(3): 297–336
- 26 Newman D J, Blake C, Merz C J. UCI repository of machine learning data bases [Online], available: <http://archive.ics.uci.edu/ml/datasets.html>, January 10, 2010.
- 27 Lo H Y, Wang J C, Wang H M, Lin H D. Cost-sensitive multi-label learning for audio tag annotation and retrieval. *IEEE Transactions on Multimedia*, 2011, **13**(3): 518–529



付忠良 中国科学院成都计算机应用研究所研究员. 主要研究方向为计算机视觉和机器学习.

E-mail: fzliang@netease.com

(FU Zhong-Liang Professor at Chengdu Institute of Computer Application, Chinese Academy of Sciences. His research interest covers computer vision and machine learning.)