一种自适应的多类Boosting分类算法

王世勋[1] 潘 鹏[2] 陈 灯[3] 卢炎生[2]
(河南师范大学计算机与信息工程学院 新乡 453007)[1]
(华中科技大学计算机科学与技术学院 武汉 430074)[2]
(武汉工程大学智能机器人湖北省重点实验室 武汉 430205)[3]

摘 要 许多实际问题涉及到多分类技术,该技术能有效地缩小用户与计算机之间的理解差异。在传统的多类Boosting方法中,多类损耗函数未必具有猜测背离性,并且多类弱学习器的结合被限制为线性的加权和。为了获得高精度的最终分类器,多类损耗函数应具有多类边缘极大化、贝叶斯一致性与猜测背离性。此外,弱学习器的缺点可能会限制线性分类器的性能,但它们的非线性结合可以提供较强的判别力。根据这两个观点,设计了一个自适应的多类Boosting分类器,即SOHPBoost算法。在每次迭代中,SOHPBoost算法能够利用向量加法或Hadamard乘积来集成最优的多类弱学习器。这个自适应的过程可以产生多类弱学习的Hadamard乘积向量和,进而挖掘出数据集的隐藏结构。实验结果表明,SOHPBoost算法可以产生较好的多分类性能。

关键词 多类Boosting,损耗函数,猜测背离性,非线性结合

中图法分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.07.033

Adaptive Multiclass Boosting Classification Algorithm

WANG Shi-xun[1] PAN Peng[2] CHEN Deng[3] LU Yan-sheng[2]
(School of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China)[1]
(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)[2]
(Hubei Provincial Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan 430205, China)[3]

Abstract Many practical problems have involved classification technology which can effectively reduce the comprehension difference between users and computers. In the traditional multiclass Boosting methods, multiclass loss function does not necessarily have guess-aversion, and the combinations of multiclass weak learners are confined to linear weighted sum. To get a final classifier with high accuracy, multiclass loss function need contain three main properties, namely margin maximization, Bayes consistency and guess-aversion. Moreover, the weakness of weak learners may limit the performance of linear classifier, but their nonlinear combinations should provide strong discrimination. Therefore, we designed an adaptive multiclass Boosting classifier, namely SOHPBoost algorithm. At every iteration, our algorithm can add the best weak learner to the current ensemble according to vector addition or Hadamard product. This adaptive process can create the sum of Hadamard products of weak learners, and mine the hidden structure of dataset. The experiments show that SOHPBoost algorithm can provide more advantageous performance of multiclass classification.

Keywords Multiclass Boosting, Loss function, Guess-aversion, Nonlinear combination

1 引言

近年来,机器学习与人工智能领域中的众多实际问题都涉及到了分类器的设计,一个值得信赖的设计工具就是串行集成的Boosting技术。它通过将多个弱学习器结合成一个强学习器,明显地提高了分类器的精度。在一般的Boosting算法中,每一个弱学习器在一个加权的数据集上训练,每一个数据的权重依据该弱学习器的性能被重新更改。当训练下一个弱学习器时,被上一个弱学习器误判的数据可以得到更多的关注。通过有限次的迭代,这些弱学习器的加权集成便形成了最终的强学习器。比较有代表性的Boosting算法包括AdaBoost[1], LogitBoost[2], Savage-Boost[3], TangentBoost[4]

到稿日期:2016-04-28 返修日期:2016-07-11 本文受河南省自然科学基金(162300410177),河南省高等学校重点科研项目(17A520040),河南师范大学博士科研启动基金(qd15134)资助。

王世勋(1985—),男,博士,讲师,主要研究方向为机器学习,E-mail:wsxun@hust.edu.cn;潘 鹏(1975—),男,博士,副教授,主要研究方向为机器学习;陈 灯(1983—),男,博士,讲师,主要研究方向为机器学习、软件工程;卢炎生(1950—),男,教授,主要研究方向为机器学习、数据库系统。

与 TaylorBoost[5] 等，它们主要用于解决两类问题。

分类学习是将数据对象分派到预先定义的某一个类别中，例如图像的场景识别与音乐的风格分类等。对于两类问题，Boosting 算法只要求每个弱学习器的分类正确率高于 50%，即大于随机猜测的概率值。对于多类问题，这样的约束条件显得有点严格，因此多类 Boosting 的设计存在一定的困难。一般而言，设计多类 Boosting 的方案可以分为间接策略和直接策略。前者是把多类问题分解成若干个两类子问题，例如“一对一”和“一对多”等。虽然间接策略在某些情况下可以取得一些成果，但它仍存在一些缺点，如引起数据分布的不平衡，增加训练的复杂度，产生不在同一级别的输出结果等[6]。在直接策略中，一组类码本通常在多类弱学习器的集成过程中起到重要的作用，但一些损耗函数的属性依然需要被考虑，例如极大化的多类边缘、贝叶斯一致性与猜测背离性。不同类别之间的误判代价往往是不同的，因此仍需对具有代价敏感性的多类 Boosting 进行进一步的研究。

在设计 Boosting 分类器的过程中，有效的损耗函数需要包括 3 个重要性质，即多类边缘极大化、贝叶斯一致性与猜测背离性。假如一些样本数据落在靠近分类边界的正确区域内，那么边缘极大化的思想是惩罚这些数据。如果分类风险的最优值逐渐收敛于贝叶斯决策规则，那么引起该风险的损耗函数便具有贝叶斯一致性。边缘的极大化可以在小数据集上提高分类器的泛化能力，而贝叶斯一致性能在大数据集上获得最优的分类器。当最终的分类器以相同的得分来评判每个类别时，猜测背离性的思想是促进正确的分类，而不是随意地猜测。虽然两类损耗函数通常具有猜测背离性，但多类损耗函数未必具有该性质[6]。除此之外，在一些复杂的问题中，弱学习器的弱判别性可能会限制最终分类器的性能。一个有效的方案是学习出比较复杂的分类器结构，即以非线性的方式集成弱学习器。

本文提出了一个代价敏感的多类逻辑损耗函数，它不仅拥有多类边缘极大化与猜测背离性，而且在代价不敏感的分类问题中具有贝叶斯一致性。依据梯度下降方法，分类风险可以在多维的凸泛函空间内降低到全局最优值。为了挖掘数据集的复杂结构，本文利用直接策略设计了一个自适应的多类 Boosting 分类器，即 SOHPBoost 算法。在每一次循环迭代中，通过自动地选择向量加法或 Hadamard 乘积，SOHPBoost 算法可以将最优的多类弱学习器集成到当前的强分类器上，进而形成弱学习器的 Hadamard 乘积向量和。本文第 2 节介绍 Boosting 分类的相关研究工作；第 3 节介绍代价敏感的多类逻辑损耗函数；第 4 节详细地介绍自适应的多类 Boosting 分类算法；第 5 节给出实验结果；最后总结全文。

2 相关工作

最早的 Boosting 算法是 AdaBoost[1]，它通过梯度下降来最小化由指数损耗引起的分类风险，进而选择每次迭代的弱学习器及其贡献系数。从统计学的角度，Fridman 等人[2]解释了 Boosting 算法的优越性，并利用 Newton 方法得到 LogitBoost 算法。Masnadi 等人[3]通过实验表明，SavageBoost 算法对异常数据点有很强的适应性。文献[4]提出了可以操纵边缘的 Tangent 损耗函数，它能对较大的正边缘和负边缘执行一定的惩罚。基于分类风险的泰勒级数展开式，Saberian 等人[5]提出了能适用于任意损耗函数与优化策略的 TaylorBoost 算法，并证明 LogitBoost 是 TaylorBoost 的特殊情况。这些方法具有一定的局限性，因为它们是以线性方式来集成弱学习器。为了提高强学习器处理类内差异的能力，Danielsson 等人[7]利用逻辑门网络结合一组已训练好的弱学习器，进而构造出门分类器。通过自动地选择预定义好的操作符，Saberian 等人[8]提出的 Boosting 框架可以产生两种新的弱学习器集成方式，即弱学习器乘积后相加与弱学习器相加后乘积。然而，这两种方法并不适用于多分类问题。

通过设计出一个二进制码本矩阵，Dietterich 等人[9]提出了 ECOC 方法。在该码本矩阵中，每行向量用来标记一个类别，而每列向量用来训练两类分类器。利用随机欠采样，Seiffert等人[10]提出了 RUSBoost 算法。通过学习一些弱学习器的乘积，Kégl 等人[11]构造了一个强学习器，并在标准数据集上证明了它拥有较好的分类性能。这 3 个多类 Boosting 算法并没有直接集成多类弱学习器，而是间接地把多类问题分解成多个两类子问题。通过给每个类分派一个码本向量，SAMME 算法[12]最小化了多类指数损耗函数的期望，并等价于一个前向的阶段性增加模型，该算法是 AdaBoost 算法在多分类问题上的自然延伸。为了识别出弱学习器所需的最优必要条件，Mukherjee 等人[13]构造了一个广泛的框架，但基于此框架的 Boosting 算法不具有贝叶斯一致性或多类边缘极大化。通过求解一个带有约束的优化问题，Saberian 等人[14]获得了一个最优的码本矩阵，并提出了两个相应的多类Boosting算法。在这些方法中，多类弱学习器的集成方式是线性的。

根据最优代价敏感学习的条件，Masnadi 等人[15]提出了一个可设计代价敏感 Boosting 算法的框架，并延伸了传统的 Boosting 算法，使其达到最优的代价敏感决策。GBSE 方法[16]的核心技术包括迭代地调整样本权重、扩展数据空间与梯度下降，它可以被用于解决多类代价敏感学习问题。通过把梯度集成的理论应用到 P-norm 损耗函数上，Lozano 等人[17]提出了一类新的代价敏感多类 Boosting 方法。基于一个简单的多类损耗函数，MultiBoost 算法[18]只要求弱学习器的分类性能优于随机猜测的性能。此外，Beijbom 等人[19]引入了猜测背离的概念，并通过实验证明了该概念是多类损耗函数的一个重要属性。

3 多类逻辑损耗函数

假设已标注的数据集被表示为 $(X,Z)=\{(\boldsymbol{x}_1,z_1),\cdots,(\boldsymbol{x}_N,z_N)\}$，其中 $z_i\in\{1,\cdots,K\}$ 是语义类的标签，N 是数据集的大小，向量 $\boldsymbol{x}_i\in\mathbb{R}^d$ 是独立地从同一个未知概率分布抽取的样本数据。令 $\boldsymbol{C}$ 表示大小为 $K\times K$ 的代价矩阵，元素 $C_{i,j}$ 是第 i 类数据被分为第 j 类的误判代价。在代价矩阵中，主对角线上的元素值均等于 0，而其他的元素值均为正值。如果任意的非主对角线元素值为 1，那么矩阵是代价不敏感的；否则，矩阵是代价敏感的。一般情况下，分类器的目标是从训练数据集中学习出一个分类规则 $F(\boldsymbol{x}):X\rightarrow\{1,\cdots,K\}$，以便新

数据 $\boldsymbol{x}$ 可获得一个语义类别标签。从最小化错误率的角度来看，最优的分类器应该执行如下的贝叶斯决策规则：

$$F(\boldsymbol{x})=\arg\min_{z}\sum_{i=1}^{K}P_{Z|X}(i|\boldsymbol{x})C_{i,z} \tag{1}$$

对于代价不敏感矩阵 $\boldsymbol{C}$，贝叶斯决策规则等价于：

$$F(x)=\arg\max_{z}P_{Z|X}(z|\boldsymbol{x}) \tag{2}$$

式(1)与式(2)中的概率值估计存在一定的困难，因此贝叶斯决策规则不容易被执行。一个可能的解决方案是求助于 Boosting 方法，它通过集成一些弱分类器来估计贝叶斯决策规则。类标签 $+1$ 和 -1 在两类问题中起着重要的作用，但它们并不能立刻延伸到 K 类问题。因此，语义类标签需要被一个多维向量 $\boldsymbol{y}$ 重新编码。为了达到该目的，文献[14]介绍了 K 个不同的单位码本集合 $Y=\{\boldsymbol{y}^1,\cdots,\boldsymbol{y}^K\}$，这些码本是中心位于原点的 $K-1$ 维正则形的顶点。经过处理，每一个类别 z 对应于一个码本 $\boldsymbol{y}^z\in\mathbb{R}^{K-1}$。

类似于两类问题中的边缘，如果 $\boldsymbol{f}(\boldsymbol{x})\in\mathbb{R}^{K-1}$ 是一个预测器，符号 $\langle\cdot,\cdot\rangle$ 表示内积，那么可将该预测器对类别标签 z 的得分定义为 $S_z(\boldsymbol{x})=\langle\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{y}^z\rangle$。得分反映出数据被分派到某一类的可信度，某类的得分值越高，数据属于该类的可能性就越大。在此基础上，多类边缘[14]为：

$$M(z,\boldsymbol{S}(\boldsymbol{x}))=[S_z(\boldsymbol{x})-\max_{l\neq z}S_l(\boldsymbol{x})]/2 \tag{3}$$

其中，$\boldsymbol{S}(\boldsymbol{x})\in\mathbb{R}^K$ 是得分向量，$S_k(\boldsymbol{x})$ 是它的第 k 个元素。为了获得概率结果，自然地执行如下的分类器：

$$F(\boldsymbol{x})=\arg\max_{z}\sigma(\langle\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{y}^z\rangle)/\sum_{z}\sigma(\langle\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{y}^z\rangle) \tag{4}$$

其中，$\sigma(\cdot)$ 是一个 Sigmoid 函数。显而易见，式(4)等价于 $F(\boldsymbol{x})=\arg\max_{z}M(z,\boldsymbol{S}(\boldsymbol{x}))$。因此，分类器 $F(\boldsymbol{x})$ 可以找到一个类别，使得该类拥有最大的多类边缘。为了获得最优的分类器，需要寻找一个最优的预测器，使得它可以最小化如下分类风险。

$$R_M(\boldsymbol{f})=E_{X,Z}\{L_M[\boldsymbol{C},z,\boldsymbol{f}(\boldsymbol{x})]\}\approx\sum_{i=1}^{N}L_M[\boldsymbol{C},z_i,\boldsymbol{f}(\boldsymbol{x}_i)] \tag{5}$$

其中，$L_M[\cdot,\cdot,\cdot]$ 是多类损耗函数。分类风险越小时，预测器的性能越好。一般而言，最优的预测器能近似地被看作多类弱学习器的线性组合。在此情况下，有如下的优化问题：

$$\min_{\boldsymbol{f}(\boldsymbol{x})}R_M[\boldsymbol{f}(\boldsymbol{x})]\quad\text{s.t. }\boldsymbol{f}(\boldsymbol{x})\in span(G) \tag{6}$$

其中，$G=\{\boldsymbol{g}_1(\boldsymbol{x}),\cdots,\boldsymbol{g}_m(\boldsymbol{x})\}$ 是多类弱学习器 $\boldsymbol{g}_i(\boldsymbol{x}):X\to\mathbb{R}^{K-1}$ 的集合，而 $span(G)$ 表示由 $\boldsymbol{g}_i(\boldsymbol{x})$ 构成的线性泛函空间。

为了获得一些对多类 Boosting 算法有益的性质，引入了代价敏感的多类逻辑损耗函数：

$$L_M[\boldsymbol{C},z,\boldsymbol{f}(\boldsymbol{x})]=\sum_{i=1}^{K}\log[1+C_{z,i}\exp(\langle\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{y}^i-\boldsymbol{y}^z\rangle)] \tag{7}$$

可以证明，式(7)的下界是 $\log[1+\exp(-2M(z,\boldsymbol{S}))]$。由于对数函数与指数函数的单调性，式(5)中的分类风险的最小化能够激励每个样本拥有较大的多类边缘，即多类逻辑损耗函数拥有边缘极大化的属性。假设数据 $\boldsymbol{x}$ 属于第 z 类的概率为 $\beta_z=P_{Z|X}(z|\boldsymbol{x})$，令分类风险关于预测器 $\boldsymbol{f}$ 的一阶导数等于 0，可得：

$$\sum_{z=1}^{K}\frac{\beta_z C_{z,k}\boldsymbol{y}^k}{C_{z,k}+\exp(\langle\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{\eta}_{z,k}\rangle)}=\sum_{i=1}^{K}\frac{\beta_k C_{k,i}\boldsymbol{y}^k}{C_{k,i}+\exp(\langle\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{\eta}_{k,i}\rangle)} \tag{8}$$

其中，$\boldsymbol{\eta}_{i,z}=\boldsymbol{y}^i-\boldsymbol{y}^z$。对于任意对称的代价矩阵 $\boldsymbol{C}$，利用如下表达式：

$$C_{z,k}+\exp(\langle\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{\eta}_{z,k}\rangle)=\sqrt{\beta_z/\beta_k} \tag{9}$$

可证明式(8)的左、右侧均等于 $\sum_{i=1}^{K}C_{k,i}\boldsymbol{y}^k\sqrt{\beta_k\beta_i}$，即最优的预测器 $\boldsymbol{f}^*(\boldsymbol{x})$ 蕴藏在式(9)中。对于代价不敏感矩阵，简化式(9)可得：

$$S_z^*(\boldsymbol{x})=\langle\boldsymbol{f}^*(\boldsymbol{x}),\boldsymbol{y}^z\rangle=\log(\sqrt{P_{Z|X}(z|\boldsymbol{x})}-c_1)+c_2 \tag{10}$$

其中，c_1 与 c_2 是常数。这表明式(4)中的分类决策等价于式(2)中的贝叶斯决策规则，因此多类逻辑损耗函数在代价不敏感的多分类问题中拥有贝叶斯一致性。

若 $\boldsymbol{f}(\boldsymbol{x})$ 属于集合 $f_z=\{\boldsymbol{f}|S_z>S_i,i\neq z\}$，则它是类 z 的支持预测器。若 $\boldsymbol{f}(\boldsymbol{x})$ 属于集合 $f_a=\{f|S_z=S_1,z=1,\cdots,K\}$，则它是随机猜测的预测器。后者对每一个类别产生相同的得分，这增加了决策规则的不确定性。通过随机地猜测，分类器可能会做出错误的分类。一个好的多类损耗函数应该鼓励更多的正确分类，并尽可能抑制随机猜测。若 $\boldsymbol{x}$ 属于第 z 类，$\boldsymbol{f}_1\in f_z$ 是任意的类 z 支持预测器，则对于任意的 $i\neq z$，$\boldsymbol{f}_1$ 导致的得分差 $S_i^1(\boldsymbol{x})-S_z^1(\boldsymbol{x})<0$。由对数函数与指数函数的单调性、代价矩阵元素的非负性可得：

$$L_M[\boldsymbol{C},z,\boldsymbol{f}_1(\boldsymbol{x})]<\sum_{i=1}^{K}\log(1+C_{z,i}) \tag{11}$$

另一方面，若 $\boldsymbol{f}_2\in f_a$ 是任意的随机猜测预测器，则 $\boldsymbol{f}_2$ 导致的得分差为 0，可得：

$$L_M[\boldsymbol{C},z,\boldsymbol{f}_2(\boldsymbol{x})]=\sum_{i=1}^{K}\log(1+C_{z,i}) \tag{12}$$

由式(11)与式(12)可知，不等式 $L_M[\boldsymbol{C},z,\boldsymbol{f}_1(\boldsymbol{x})]<L_M[\boldsymbol{C},z,\boldsymbol{f}_2(\boldsymbol{x})]$ 成立。因此，根据猜测背离性的定义[19]，式(7)中的多类逻辑损耗函数具有猜测背离性。

4　自适应的多类 Boosting 算法

一般而言，具有如上 3 个性质的多类损耗函数可以大幅度地提高最终分类器的性能。尽管如此，为了保证最优解可以收敛于全局最优的预测器，凸优化问题必须满足两个条件，即凸的解空间与凸的目标函数。

4.1　线性空间的凸优化

显然，式(6)中的泛函空间 $span(G)$ 是一个凸集。如果沿用前文的标记符号，那么分类风险关于预测器 $\boldsymbol{f}$ 的二阶导数为：

$$\frac{\partial^2 R_M(\boldsymbol{f})}{\partial\boldsymbol{f}(\boldsymbol{x})^2}=\sum_{z=1}^{K}\sum_{i=1}^{K}\beta_z C_{z,i}[\boldsymbol{\eta}_{i,z}\boldsymbol{\eta}_{i,z}^{\mathrm{T}}]\frac{\exp(\langle\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{\eta}_{i,z}\rangle)}{[1+C_{z,i}\exp(\langle\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{\eta}_{i,z}\rangle)]^2} \tag{13}$$

由文献[14]可知，所有的单位码本向量是不同的，即对于任意不同的 i 与 z，不等式 $\boldsymbol{\eta}_{i,z}\neq 0$ 成立。根据矩阵的理论知识，矩阵 $[\boldsymbol{\eta}_{i,z}\boldsymbol{\eta}^{\mathrm{T}}{}_{i,z}]$ 是正定矩阵。此外，代价矩阵的元素 $C_{z,i}$ 与概率 β_z 均为非负值，且存在类别 i 与 z，使得它们的值均大于 0。可以看出，分类风险的二阶导数是正定矩阵的和，因此它本身也是一个严格的正定矩阵，即式(5)中的分类风险是凸函数。综上所述，式(6)中的优化问题具有全局最优解。

假设 $\boldsymbol{f}^t(\boldsymbol{x})$ 表示第 t 次集成迭代后的预测器估计值，那么

在该估计值的邻近区域内，沿着多类弱学习器 $\boldsymbol{g}(\boldsymbol{x})$ 的方向，分类风险 $R_M(\boldsymbol{f})$ 的一阶泛函导数为：

$$\delta R_M[\boldsymbol{f}^t;\boldsymbol{g}]=\frac{\partial R_M[\boldsymbol{f}^t+\varepsilon\boldsymbol{g}]}{\partial\varepsilon}|_{\varepsilon=0}=-\sum_{i=1}^{N}\langle\boldsymbol{g}(\boldsymbol{x}_i),\boldsymbol{w}_i\rangle \quad (14)$$

其中：

$$\boldsymbol{w}_i=\sum_{k=1}^{K}(\boldsymbol{y}^{z_i}-\boldsymbol{y}^k)\frac{C_{z_i,k}\exp(\langle\boldsymbol{f}^t(\boldsymbol{x}_i),\boldsymbol{y}^k-\boldsymbol{y}^{z_i}\rangle)}{1+C_{z_i,k}\exp(\langle\boldsymbol{f}^t(\boldsymbol{x}_i),\boldsymbol{y}^k-\boldsymbol{y}^{z_i}\rangle)} \quad (15)$$

在第 $t+1$ 次集成迭代的过程中，根据梯度下降方法，最大限度地减小分类风险的多类弱学习器 $\boldsymbol{g}^*(\boldsymbol{x})$ 可表示为：

$$\boldsymbol{g}^*=\arg\min_{\boldsymbol{g}\in G}\delta R_M[\boldsymbol{f}^t;\boldsymbol{g}] \quad (16)$$

4.2 非线性空间的凸优化

在目前的多类 Boosting 方法中，弱学习器的线性结合不能充分地学习到广泛的判别力度，进而限制了最终分类器的精确性。为了解决这个问题，需要进一步研究更加复杂的多类弱学习器结合方式，从而产生拥有较强判别力的最终分类器。

若多类弱学习器的非线性结合方式是 Hadamard 乘积的向量和，则相应的非线性泛函空间表示为：

$$\Omega_G=\{\boldsymbol{h}(\boldsymbol{x})\mid\boldsymbol{h}(\boldsymbol{x})=\sum_j\boldsymbol{g}_{j,1}(\boldsymbol{x})\odot\cdots\odot\boldsymbol{g}_{j,m}(\boldsymbol{x}),\boldsymbol{g}\in G\} \quad (17)$$

其中，$\odot$ 表示 Hadamard 乘积。假设 $\boldsymbol{h}_1$ 与 $\boldsymbol{h}_2$ 是空间 Ω_G 内的元素，则可以证明 $\boldsymbol{h}_1+\boldsymbol{h}_2$ 仍然属于该空间。因此，非线性泛函空间 Ω_G 是一个凸集。已知式(5)中的分类风险是凸函数，那么优化问题：

$$\min_{\boldsymbol{f}(\cdot)}R_M[\boldsymbol{f}(\boldsymbol{x})]\quad \text{s.t. } \boldsymbol{f}(\boldsymbol{x})\in\Omega_G \quad (18)$$

可产生一个全局的最优值。具体地，假设第 t 次集成迭代后的预测器估计值可定义为 $\boldsymbol{f}^t(\boldsymbol{x})=\sum_{j=1}^{U}\boldsymbol{p}_j^t(\boldsymbol{x})$，其中 U 表示加法操作符的数目，而第 j 个加法项 $\boldsymbol{p}_j^t(\boldsymbol{x})$ 表示多类弱学习器的 Hadamard 乘积，即：

$$\boldsymbol{p}_j^t(\boldsymbol{x})=\boldsymbol{g}_{j,1}(\boldsymbol{x})\odot\cdots\odot\boldsymbol{g}_{j,m^j}(\boldsymbol{x}),m^j\in\mathbb{N} \quad (19)$$

其中，m^j 是 Hadamard 乘积项的数目。需要注意的是，加法项 $\boldsymbol{p}_j^t(\boldsymbol{x})$ 可以是单个的多类弱学习器，即 $m^j=1$；$\boldsymbol{f}^t(\boldsymbol{x})$ 也可以是单个的加法项，即 $U=1$。

在第 $t+1$ 次集成迭代的过程中，本文利用向量加法与 Hadamard 乘积来更新第 t 次迭代后的预测器 $\boldsymbol{f}^t(\boldsymbol{x})$。在向量加法更新的情况下，一个最优的多类弱学习器被添加到当前的预测器 $\boldsymbol{f}^t(\boldsymbol{x})$ 中，此更新过程是标准的线性多类 Boosting 方法。根据梯度下降的优化策略，最优的多类弱学习器可由式(16)求解。若 $\boldsymbol{g}_0^*$ 是向量加法更新的最优多类弱学习器，则最优的步长可表示为：

$$\alpha_0^*=\arg\min_{\alpha\in\mathbb{R}}R_M(\boldsymbol{f}^t+\alpha\boldsymbol{g}_0^*) \quad (20)$$

向量加法更新后的预测器可以导致如下的分类风险：

$$R_M^0(\boldsymbol{f}^{t+1})=R_M(\boldsymbol{f}^t+\alpha_0^*\boldsymbol{g}_0^*) \quad (21)$$

在 Hadamard 乘积更新的情况下，本文利用一个新的多类弱学习器 $\boldsymbol{g}(\boldsymbol{x})$ 来更新每个加法项。例如，第 j 个加法项在 Hadamard 乘积更新后变成 $\boldsymbol{p}_j^{t+1}(\boldsymbol{x})=\boldsymbol{p}_j^t(\boldsymbol{x})\odot\boldsymbol{g}(\boldsymbol{x})$。显而易见，更新后的预测器为：

$$\begin{aligned}\boldsymbol{f}^{t+1}(\boldsymbol{f})&=\sum_{i\neq j}\boldsymbol{p}_i^t(\boldsymbol{x})+\boldsymbol{p}_j^t(\boldsymbol{x})\odot\boldsymbol{g}(\boldsymbol{x})\\&=\boldsymbol{Q}_j^t(\boldsymbol{x})+\boldsymbol{p}_j^t(\boldsymbol{x})\odot\boldsymbol{g}(\boldsymbol{x})\end{aligned} \quad (22)$$

其中，$\boldsymbol{Q}_j^t(\boldsymbol{x})=\boldsymbol{f}^t(\boldsymbol{x})-\boldsymbol{p}_j^t(\boldsymbol{x})$。对于式(22)中的更新，分类风险 $R_M(\boldsymbol{f})$ 在 $\boldsymbol{Q}_j^t(\boldsymbol{x})$ 的邻近区域内的一阶泛函导数为：

$$\begin{aligned}\delta R_M[\boldsymbol{f}^t;\boldsymbol{g},j]&=\frac{\partial R_M[\boldsymbol{Q}_j^t+\varepsilon\boldsymbol{p}_j^t\odot\boldsymbol{g}]}{\partial\varepsilon}|_{\varepsilon=0}\\&=-\sum_{i=1}^{N}\langle\boldsymbol{g}(\boldsymbol{x}_i),\boldsymbol{\varphi}_i\rangle\end{aligned} \quad (23)$$

其中：

$$\boldsymbol{\varphi}_i=\sum_{k=1}^{K}\boldsymbol{p}_j^t(\boldsymbol{x}_i)\odot(\boldsymbol{y}^{z_i}-\boldsymbol{y}^k)\frac{C_{z_i,k}\exp(\langle\boldsymbol{Q}_j^t(\boldsymbol{x}_i),\boldsymbol{y}^k-\boldsymbol{y}^{z_i}\rangle)}{1+C_{z_i,k}\exp(\langle\boldsymbol{Q}_j^t(\boldsymbol{x}_i),\boldsymbol{y}^k-\boldsymbol{y}^{z_i}\rangle)} \quad (24)$$

根据式(16)与式(23)，可以获得最大限度地减小分类风险 $R_M(\boldsymbol{f}^{t+1})$ 的多类弱学习器。若 $\boldsymbol{g}_j^*$ 是第 j 个加法项中 Hadamard 乘积更新的最优多类弱学习器，则沿着该方向的最优步长为：

$$\alpha_j^*=\arg\min_{\alpha\in\mathbb{R}}R_M(\boldsymbol{Q}_j^t+\alpha\boldsymbol{p}_j^t\odot\boldsymbol{g}_j^*) \quad (25)$$

因此，Hadamard 乘积更新后的预测器导致的分类风险为：

$$R_M^j(\boldsymbol{f}^{t+1})=R_M(\boldsymbol{Q}_j^t+\alpha_j^*\boldsymbol{p}_j^t\odot\boldsymbol{g}_j^*) \quad (26)$$

4.3 非线性集成的多类 Boosting 算法

综上讨论，对于一个向量加法更新与 U 个 Hadamard 乘积更新，自适应的多类 Boosting 方法首先计算每个更新操作的最优弱学习器及其相应的分类风险；然后选择一个最佳的更新操作，使得该更新操作能够最大限度地减小分类风险。为了叙述方便，自适应的多类 Boosting 算法被命名为 SOHP-Boost，它主要包括 5 个步骤。

步骤 1 输入数据集 (X,Z)、类别数目 K、码本向量集合 Y、代价矩阵 $\boldsymbol{C}$、多类损耗函数 L_M 与迭代次数 T；当迭代循环开始时，设置 $U=0$，$\boldsymbol{f}^0(\boldsymbol{x})=\boldsymbol{0}\in\mathbb{R}^{K-1}$。

步骤 2 在第 $t\in\{0,\cdots,T-1\}$ 次迭代循环中，对于向量加法更新，先用式(14)与式(16)寻找最优的多类弱学习器 $\boldsymbol{g}_0^*$，再用式(20)寻找最优的步长 α_0^*，最后利用式(21)计算由向量加法更新的预测器所导致的分类风险 R_M^0。

步骤 3 对于第 $j\in\{1,\cdots,U\}$ 个加法项，先利用式(23)与式(16)寻找最优的弱学习器 $\boldsymbol{g}_j^*$，再利用式(25)寻找最优的步长 α_j^*，最后利用式(26)计算由 Hadamard 乘积更新的预测器所导致的分类风险 R_M^j。

步骤 4 令 $j_*=\arg\min_j R_M^j, j\in\{0,\cdots,U\}$，如果 j_* 的值为 0，那么 $\boldsymbol{p}_{U+1}^{t+1}$ 的值为 $\alpha_0^*\boldsymbol{g}_0^*$，且 U 的值增 1；否则 $\boldsymbol{p}_{j_*}^{t+1}$ 的值为 $\alpha_{j_*}^*\boldsymbol{p}_{j_*}^t\odot\boldsymbol{g}_{j_*}^*$。对于任意的 $j\neq j_*$，令 $\boldsymbol{p}_j^{t+1}$ 的值为 $\boldsymbol{p}_j^t$。

步骤 5 更新预测器 $\boldsymbol{f}^{t+1}(\boldsymbol{x})$ 的值为 $\sum_{j=1}^{U}\boldsymbol{p}_j^{t+1}(\boldsymbol{x})$，且令 t 的值增 1。返回步骤 2 继续执行迭代循环，直到循环条件不满足。迭代循环结束时，输出最终的预测器 $\boldsymbol{f}^T(\boldsymbol{x})$。

给出一个新数据 $\boldsymbol{x}$，通过将 SOHPBoost 输出的最终预测器 $\boldsymbol{f}^T(\boldsymbol{x})$ 代入式(4)中，可以获得该数据的分类规则。SOHPBoost 不是提前设置好向量加法与 Hadamard 乘积的操作符数目，而是从训练集中自适应地学习出这些操作符数目。在每一次集成迭代的过程中，SOHPBoost 首先计算由一个向量加法更新与若干个 Hadamard 乘积更新所引起的风险；然

后通过比较这些风险，选择出最优的更新，并把相应的多类弱学习器集成到当前的预测器上。对于一个多分类数据集，上述的学习过程可以自动地发现多类弱学习器的最优集成。因此，SOHPBoost 有可能捕获该数据集的隐藏结构。

在泛函空间 Ω_G 中，SOHPBoost 利用梯度下降方法非线性地集成了多类弱学习器。理论上，SOHPBoost 可以在空间 Ω_G 内实现两个特殊的集成，即 Hadamard 乘积集成与向量加法集成。前者仅利用 Hadamard 乘积将多类弱学习器集成在一起，该集成方式在一定程度上是文献[11]在直接多分类问题上的一个拓展。后者在集成多类弱学习器时仅依赖于向量加法操作符，这种集成方式在结构形式上类似于文献[14]。

SOHPBoost 包含了内循环与外循环，即 Hadamard 乘积更新的循环与迭代集成的循环。内循环的主要作用是寻找所有最优的 Hadamard 乘积更新，外循环在每一次迭代中均计算最优的向量加法更新。对于每一次循环，多类弱学习器的计算占据了主要的时间开销。假设多类弱学习器的计算代价为 $O(\mu)$。在 Hadamard 乘积集成的情况下，内、外循环一共计算了 $2T-1$ 次多类弱学习器，此时 SOHPBoost 获得最好的时间复杂度 $O(\mu T)$。在向量加法集成的情况下，内、外循环一共计算了 $\frac{T(T+1)}{2}$ 次多类弱学习器，此时 SOHPBoost 获得最坏的时间复杂度 $O(\mu T^2)$。在实际应用中，该算法的时间复杂度依赖于具体的数据集。

5　实验结果与分析

本文从 UCI 机器学习库[20]中下载了 10 个数据集，如表 1 所列。对于代价不敏感的多分类问题，在前 5 个数据集上比较了 SOHPBoost 算法与其他方法的性能。采用分类准确率作为评估度量标准，准确率越高，则分类器的性能就越好。对于代价敏感的多分类问题，在后 5 个数据集上比较了 SOHPBoost 与存在的多类 Boosting 的性能。采用如下的分类风险作为评估度量标准：

$$Risk=\frac{1}{N}\sum_{i=1}^{N}C_{z_i,F(x_i)} \tag{27}$$

其中，$F(\cdot)$ 表示最终的分类决策规则。分类风险越小，则最终分类器的性能越好。

表 1　10 个 UCI 数据集的概况

数据集名称	数据集数目	测试集数目	类别数目	属性数目
Isolet	7797	1559	26	617
Landsat	6435	2000	6	36
Letter	20000	4000	26	16
Pendigit	10992	3498	10	16
Shuttle	58000	14500	7	9
Breast	106	25	6	9
Ecoli	336	78	8	7
Libras	360	68	15	90
Optical	5620	1797	10	64
Vertebral	310	71	3	6

给出一对类别标签，人们通常依据类别距离对它们赋予对称的误分类代价。在代价敏感的多分类实验中，每个数据集的对称代价矩阵是随机生成的。除了主对角线上的元素，余下的元素服从[1,10]上的均匀分布。为了减小由随机性带来的误差，对每个数据集重复地生成 10 个对称的代价矩阵。根据自顶向下的递归策略，决策树可以应用于直接的多分类问题。在本文的所有实验中，深度为 2 的决策树充当了多类弱学习器的角色。一般而言，决策树的输出是一个类别数字。因此，在执行多分类算法的过程中，需要将类别数字的输出翻译成相应的类码本向量。除此之外，为了构造一个公平的对比环境，所有算法的迭代循环次数都被设定为 50。

5.1　代价不敏感的多分类

为了验证 SOHPBoost 算法的有效性，表 2 列出了该算法与其他多类 Boosting 算法的分类准确率，其中黑体数值表示最优的性能。在解决多分类问题时，LPBoost 方法与 TotalBoost 方法分别利用线性规划与二次规划来最大化数据集的极小边缘[21]。从表 2 可以看出，SOHPBoost 算法在所有数据集上获得了最优的分类准确率。例如，在 Letter 数据集上，分类准确率从先前最好的 58.8%提高到 62.08%。与 GD-MCBoost 方法相比，SOHPBoost 算法可以在一个非线性泛函空间中寻找最优的多类弱学习器，这有助于最终的分类器从数据集中挖掘出一些隐藏的结构。另一方面，多类弱学习器不会集成为一个复杂的预测器，除非该预测器可以带来较好的多分类性能。从这两个方面来说，SOHPBoost 算法优于 GD-MCBoost 方法。与表 2 中的其他方法相比，SOHPBoost 算法除了具有上述两优点之外，还存在以下 4 个优点：1)贝叶斯一致性；2)多类边缘的极大化；3)猜测背离性；4)分类风险的凸优化。最后，图 1 示出了 SOHPBoost 算法与 GD-MCBoost 方法在 Letter 数据集上的分类准确率。

表 2　不同的多类 Boosting 方法在 5 个 UCI 数据集上的分类准确率/%

算法	Isolet	Landsat	Letter	Pendigit	Shuttle	Average
AdaBoost-M2[1]	19.24	78.90	16.33	64.92	93.91	54.66
LPBoost[21]	11.55	71.60	9.83	45.68	32.59	34.25
TotalBoost[21]	30.21	73.65	22.93	76.13	74.02	55.39
RUSBoost[10]	11.55	44.95	9.83	29.27	79.16	34.95
AdaBoost-SAMME[12]	61.00	79.80	45.65	83.82	99.70	73.99
AdaBoost-Cost[13]	63.69	83.95	42.00	80.53	99.55	73.94
GD-MCBoost[14]	84.28	86.35	58.80	92.94	99.73	84.82
SOHPBoost	**85.82**	**87.15**	**62.08**	**94.03**	**99.97**	**85.81**

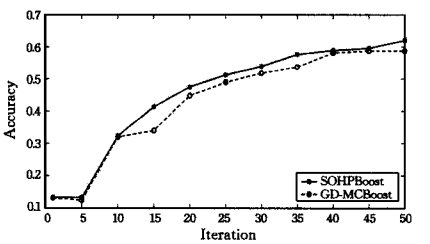

图 1　SOHPBoost 与 GD-MCBoost 在 Letter 上的分类准确率

5.2　代价敏感的多分类

给出任意的对称代价矩阵，文献[19]在理论上证明了代价敏感的 GD-MCBoost 方法的分类性能不依赖于该代价矩阵，即 GD-MCBoost 方法在代价敏感与代价不敏感两种情况下可以执行等价的分类决策。因此，本实验采用了代价不敏

感的 GD-MCBoost 方法，但这并不影响实验的执行。图 2 示出了 SOHPBoost 算法与 GD-MCBoost 方法在真实数集上的性能评估。可以看出，SOHPBoost 在所有的数据集上获得了较小的平均分类风险。例如，与 GD-MCBoost 方法相比，SOHPBoost 算法在 Libras 数据集上的分类风险大约减少了 20%，其值约为 1.19。因此，代价敏感矩阵有利于提高 SOHPBoost 算法的分类性能。产生这种现象的原因在于 SOHPBoost 算法是以非线性的方式来自适应地集成多类弱学习器。一般情况下，因 SOHPBoost 算法是在较大的搜索空间中来寻找最终的分类器，故它具有较大的时间开销。

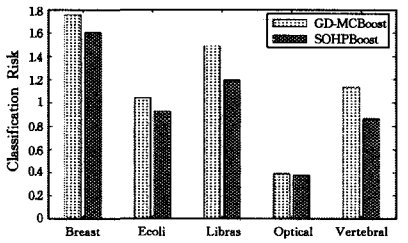

图 2 不同的代价敏感多分类方法在 5 个数据集上的平均分类风险

结束语 自适应的多分类方法 SOHPBoost 可以直接从非线性泛函空间中学习出最终的分类器。虽然 SOHPBoost 算法是一些 Boosting 方法的变体，但其与已有方法存在如下不同之处：1)一个新的代价敏感的多类逻辑损耗函数；2)多类弱学习器的非线性集成，即 Hadamard 乘积向量和。除此之外，SOHPBoost 算法还具有一些优点：1)代价不敏感的贝叶斯一致性；2)多类边缘极大化；3)猜测背离性；4)分类风险的凸优化。大量的实验结果表明，SOHPBoost 算法可以产生较好的分类性能。

参 考 文 献

[1] FREUND Y, SCHAPIRE R E. A Decision-theoretic Generalization of On-line Learning and an Application to Boosting[J]. Journal of Computer and System Sciences, 1997, 55(1): 119-139.

[2] FRIEDMAN J, HASTIE T, TIBSHIRANI R. Additive Logistic Regression: A Statistical View of Boosting[J]. The Annals of Statistics, 2000, 28(2): 337-407.

[3] MASNADI-SHIRAZI H, VASCONCELOS N. On the Design of Loss Functions for Classification: Theory, Robustness to Outliers, and Savagebooost[C]// Proceedings of Advances in Neural Information Processing Systems. 2009: 1049-1056.

[4] MASNADI-SHIRAZI H, MAHADEVAN V, VASCONCELOS N. On the Design of Robust Classifiers for Computer Vision [C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2010: 779-786.

[5] SABERIAN M J, MASNADI-SHIRAZI H, VASCONCELOS N. TaylorBoost: First and Second-order Boosting Algorithms with Explicit Margin Control[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2011: 2929-2934.

[6] ABOUELENIEN M, YUAN X. Boosting for Learning from Multiclass Data Sets via a Regularized Loss Function[C]// Proceedings of IEEE International Conference on Granular Computing. 2013: 4-9.

[7] DANIELSSON O, RASOLZADEH B, CARLSSON S. Gated Classifiers: Boosting under High Intra-Class Variation[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2011: 2673-2680.

[8] SABERIAN M J, VASCONCELOS N. Boosting Algorithms for Simultaneous Feature Extraction and Selection[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2012: 2448-2455.

[9] DIETTERICH T G, BAKIRI G. Solving Multiclass Learning Problems via Error Correcting Output Codes[J]. Journal of Artifical Intelligence Research, 1995, 2: 263-286.

[10] SEIFFERT C, KHOSHGOFTAAR T M, HULSE V, et al. RUSBoost: Improving Classification Performance When Training Data Is Skewed[C]// Proceedings of IEEE International Conference on Pattern Recognition. 2008: 1-4.

[11] KÉGL B, BUSA-FEKETE R. Boosting Products of Base Classifiers[C]// Proceedings of the 26th International Conference on Machine Learning. 2009: 497-504.

[12] ZHU J, ZOU H, ROSSET S, et al. Multi-class AdaBoost[J]. Statistics and Its Interface, 2009, 2(3): 349-360.

[13] MUKHERJEE I, SCHAPIRE R E. A Theory of Multiclass Boosting[J]. Journal of Machine Learning Research, 2013, 14(1): 437-497.

[14] SABERIAN M J, VASCONCELOS N. Multiclass Boosting: Theory and Algorithms [C] // Proceedings of Advances in Neural Information Processing Systems. 2011: 2124-2132.

[15] MASNADI-SHIRAZI H, VASCONCELOS N. Cost-sensitive Boosting[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(2): 294-309.

[16] ABE N, ZADROZNY B, LANGFORD J. An Iterative Method for Multi-class Cost-sensitive Learning[C]// Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2004: 3-11.

[17] LOZANO A C, ABE N. Multi-class Cost-sensitive Boosting with P-norm Loss Functions[C] // Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2008: 506-514.

[18] WANG J. Boosting the Generalized Margin in Costsensitive Multiclass Classification[J]. Journal of Computational and Graphical Statistics, 2013, 22(1): 178-192.

[19] BEIJBOM O, SABERIAN M, KRIEGMAN D, et al. Guess-Averse Loss Functions for Cost-sensitive Multiclass Boosting[C]// Proceedings of the 31st International Conference on Machine Learning. 2014: 586-594.

[20] UCI: Machine Learning Repository[OL]. http://archive.ics.uci.edu/ml/datasets.html.

[21] WARMUTH M K, LIAO J, RÄTSCH G. Totally Corrective Boosting Algorithm That Maximize the Margin[C] // Proceedings of 23rd ACM International Conference on Machine Learning. 2006: 1001-1008.