组合凸线性感知器的极大切割构造方法

冷强奎 [1] 李玉鑑 [1]

摘 要 组合凸线性感知器 (Multiconlitron) 是用来构造分片线性分类器的一个通用理论框架, 对于凸可分和叠可分情况, 分别使用支持凸线性感知器算法 (Support conlitron algorithm, SCA) 和支持组合凸线性感知器算法 (Support multiconlitron algorithm, SMA) 将两类样本分开. 本文在此基础上, 提出了一种基于极大切割 (Maximal cutting) 的组合凸线性感知器构造方法. 该方法由两阶段训练构成, 第一阶段称为极大切割过程 (Maximal cutting process, MCP), 通过迭代不断寻求能够切开最多样本的线性边界, 并因此来构造尽可能小的决策函数集, 最大程度减少决策函数集中线性函数的数量, 最终简化分类模型. 第二阶段称为边界调整过程 (Boundary adjusting process, BAP), 对 MCP 得到的初始分类边界进行一个二次训练, 调整边界到适当位置, 以提高感知器的泛化能力. 数值实验说明, 此方法能够产生更为合理的分类模型, 提高了感知器的性能. 同其他典型分片线性分类器的性能对比, 也说明了这种方法的有效性和竞争力.

关键词 组合凸线性感知器, 极大切割, 两阶段训练, 泛化能力, 分片线性分类器

引用格式 冷强奎, 李玉鑑. 组合凸线性感知器的极大切割构造方法. 自动化学报, 2014, **40**(4): 721−730

DOI 10.3724/SP.J.1004.2014.00721

Maximal Cutting Construction for Multiconlitron

LENG Qiang-Kui[1] LI Yu-Jian[1]

Abstract Multiconlitron is a general framework for constructing piecewise linear classifiers. For convexly separable and commonly separable data sets, it can separate them correctly by using support conlitron algorithm (SCA) and support multiconlitron algorithm (SMA), respectively. On this basis, the paper proposes a maximal cutting construction method for multiconlitron design. The method consists of two training processes. In the first step, the maximal cutting process (MCP) is utilized iteratively to find a linear boundary such that it can obtain the maximum number of samples. Thus, the MCP can reduce the number of linear boundaries and construct a minimal set of decision functions, and ultimately simplify the classification model. To improve the generalization ability further, in the second step we employ a boundary adjusting process (BAP) to make the classification boundaries more fittable. Experiments on both synthetic and real data sets show that the presented method can produce more reasonable multiconlitron with better performance. Comparison with some other piecewise linear classifiers verifies its effectiveness and competitiveness.

Key words Multiconlitron, maximal cutting, two-step training, generalization ability, piecewise linear classifier

Citation Leng Qiang-Kui, Li Yu-Jian. Maximal cutting construction for multiconlitron. *Acta Automatica Sinica*, 2014, **40**(4): 721−730

分片线性学习的核心是构造分片线性分类器, 这是一项具有挑战性并且复杂的任务, 是模式识别领域中的一个基本问题. 由于分片线性分类器确定的决策面由若干个超平面段组成, 所以与一般超曲面相比, 仍然是简单易于实现的, 且需要较少的内存消耗. 同时由于它是由多段超平面组成的, 所以它能逼近各种形状的超曲面, 具有很强的适应能力[1].

由于上述令人熟知的优点, 分片线性分类器已经引起了很大关注, 许多设计分片线性分类器的方法也被提出来. Nilsson 的 Committee machine[2] 可看作分片线性分类器的特殊形式, 但需要进行复杂的判别步骤并花费巨大的计算代价. Mangasarian[3] 利用线性规划算法来构造分片线性分类器, Herman 等[4] 后来提出了一种相似但更好的策略, 即在构造过程中使用线性异常函数 (Linear abnormality functions). 然而上述两个方法需要执行线性规划多次, 代价较高.

在 1980 年, Sklansky 等[5] 提出一种局部训练的方法来设计分片线性分类器. 首先使用 Forgy 聚类

收稿日期 2012-10-12 录用日期 2013-10-25
Manuscript received October 12, 2012; accepted October 25, 2013
国家自然科学基金 (61175004), 北京市自然科学基金 (4112009), 北京市教委科技发展重点项目 (KZ201210005007), 高等学校博士学科点专项科研基金 (20121103110029) 资助
Supported by National Natural Science Foundation of China (61175004), Natural Science Foundation of Beijing (4112009), Beijing Municipal Education Commission Science and Technology Development Plan Key Project (KZ201210005007), and Specialized Research Fund for the Doctoral Program of Higher Education (20121103110029)
本文责任编委 张长水
Recommended by Associate Editor ZHANG Chang-Shui
1. 北京工业大学计算机学院 北京 100124
1. College of Computer Science, Beijing University of Technology, Beijing 100124

算法形成原型, 然后找到原型间的紧互对并进行局部训练, 以此产生初始的超平面. 基于局部训练的思想, Park 等[6] 提出了一种切 Tomek 链的算法来设计多类的分片线性分类器. 但这种方法常常由于超平面不够充分而遭受欠适的影响. 为了解决这个问题, Tenmoto 等[7] 使用了最小描述长度准则 (Minimum description length, MDL) 来选择合适数量的超平面, 并且将局部训练错误率保持在特定阈值之下.

决策树是构造分片线性分类器的另一种策略. 在 1996 年, Cai 等[8] 使用二叉树结构和遗传算法, 在最大化降噪准则 (Maximum impurity reduction criterion) 的指导下来设计分片线性分类器. 在 2006 年, Kostin[9] 设计并实现了一种简单快速的多类分类算法, 并且保证了精度要求.

2011 年, Li 等[10] 在凸可分、凸线性感知器和组合凸线性感知器的概念基础上, 提出了一个构造分片线性分类器的通用理论框架. 在这个框架下, 交叉距离最小化算法 (Cross distance minimization algorithm, CDMA) 用来解决线性可分问题. 支持凸线性感知器算法 (Support conlitron algorithm, SCA) 用来解决凸可分问题, 支持组合凸线性感知器算法 (Support multiconlitron algorithm, SMA) 用来解决叠可分问题. 从理论上来说, SCA 和 SMA 可看作支持向量机 (Support vector machines, SVMs) 的无核推广. 它们的性能一般优于线性核 SVM, 但低于径向基函数 (Radiad basis function, RBF) 核 SVM. 提高 SCA 和 SMA 的性能成为新的研究问题.

在本文中, 我们提出了一种极大切割 (Maximal cutting) 的方法来改进 SCA 和 SMA. 通过两阶段训练过程, 使它们产生更少的决策函数数量, 以得到更简化的分类模型, 同时调整线性边界到更加合理的位置, 增加其泛化能力.

全文组织结构如下: 第 1 节介绍 CDMA 算法, 它是本文方法的基础组成部分; 第 2 节和第 3 节分别给出基于极大切割的 SCA 算法和 SMA 算法; 第 4 节为数值实验; 最后是总结.

1 交叉距离最小化算法

在研究凸包和 SVMs 之间关系的基础上, CDMA 算法被提出[10], 同时它也作为后续算法的基础组成部分. 如果 $X, Y \subseteq \mathbf{R}^n$ 是两个有限集, 并且是线性可分的, 通过 CDMA 算法能够计算得到 X 的凸包和 Y 的凸包之间的最近点, 并由此构造一个硬间隔的 SVM, 将两类样本正确分开.

令 $CH(S)$ 表示 S 的凸包. 如果 X 和 Y 是线性可分的, 求解 SVM 可被相应地转换为求两个凸包 $CH(X)$ 和 $CH(Y)$ 之间的最近点问题[11]:

$$\begin{aligned} &\min \|x-y\| \\ &\text{s.t.} \quad x \in CH(X),\ y \in CH(Y) \end{aligned} \tag{1}$$

假设 (x^*, y^*) 是经过式 (1) 计算得到的一个最近点对, 即 $d(CH(X), CH(Y)) = \|x^* - y^*\|$, 我们可以得到它们的垂直平分面就是集合 X 和 Y 的硬间隔 SVM, 分界函数可被表示为

$$f(x) = w^* \cdot x + b^* \tag{2}$$

其中, $w^* = x^* - y^*$, $b^* = [(y^* - x^*) \cdot (y^* + x^*)]/2 = (\|y^*\|^2 - \|x^*\|^2)/2$, 分类间隔定义为

$$margin(f) = \|w^*\| = \|x^* - y^*\| \tag{3}$$

CDMA 算法描述如算法 1 所示, 算法中的第 2 (b) 步和第 2 (c) 步, 目的是在每次迭代中寻找更近的点对 (x^*, y^*), 图 1 给出了找更近点对的几何解释. 如果 x_1 不是 $CH(X)$ 距离 y^* 的最近点, 一定存在另一个点 $z^* = x_2$ 或 $z^* = x_1 + \lambda(x_2 - x_1)$, 使得 $d(z^*, y^*) < d(x_1, y^*)$. 相应地, 当第 2 (b) 步执行完毕后, 第 2 (c) 步找到距离 x^* 最近的 y^*, 直到满足最终的收敛条件 $d(x_1, y_1) - d(x^* - y^*) < \varepsilon$.

算法 1. 交叉距离最小化算法 (CDMA)

Algorithm CDMA (X, Y, ε)

1) $x^* \in X$, $y^* \in Y$;

2) Repeat

(a) $x_1 = x^*$, $y_1 = y^*$;

(b) $x^* = \arg\min\limits_{z} \left\{ d(z, y^*) \,\middle|\, \begin{array}{l} z = x_2, \text{ if } \lambda \geq 1 \\ z = x_1 + \lambda(x_2 - x_1), \\ \qquad \text{if } 0 < \lambda < 1 \end{array} \right.$

$\left. \lambda = \frac{(x_2 - x_1)(y^* - x_1)}{(x_2 - x_1)(x_2 - x_1)} > 0, x_2 \neq x_1, x_2 \in X \right\}$;

(c) $y^* = \arg\min\limits_{z} \left\{ d(x^*, z) \,\middle|\, \begin{array}{l} z = y_2, \text{ if } \lambda \geq 1 \\ z = y_1 + \lambda(y_2 - y_1), \\ \qquad \text{if } 0 < \lambda < 1 \end{array} \right.$

$\left. \lambda = \frac{(y_2 - y_1)(x^* - y_1)}{(y_2 - y_1)(y_2 - y_1)} > 0, y_2 \neq y_1, y_2 \in Y \right\}$;

until $d(x_1, y_1) - d(x^* - y^*) < \varepsilon$

3) $w^* = x^* - y^*$, $b^* = (\|y^*\|^2 - \|x^*\|^2)/2$;

4) Return $f(x) = w^* \cdot x + b^*$.

令 $|S|$ 表示集合 S 的基数. CDMA 的时间复杂度为 $\mathrm{O}(D(|X| + |Y|)/\varepsilon)$, 其中 ε 称为精度参数, 用于控制收敛条件, $D = \max_{x \in X, y \in Y}\{\|x - y\|\}$, D/ε 的值与样本分布有关, 表示在 ε 精度下, 算法收敛的最大迭代次数.

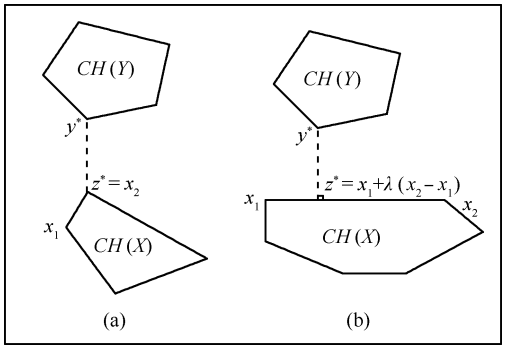


图 1　z^* 的几何意义 ((a) 如果 $\lambda \geq 1$, 则 $z^* = x_2$ 是 X 中的一个点; (b) 如果 $0 < \lambda < 1$, 则 $z^* = x_1 + \lambda(x_2 - x_1)$ 为 y^* 到线段 $CH\{x_1, x_2\}$ 的垂点)

Fig. 1　The geometric explanation of z^* ((a) If $\lambda \geq 1$, then $z^* = x_2$ is a point in X; (b) If $0 < \lambda < 1$, then $z^* = x_1 + \lambda(x_2 - x_1)$ is actually the vertical point from y^* to the line segment $CH\{x_1, x_2\}$.)

2　极大切割的支持凸线性感知器算法

CDMA 算法通过计算凸包间最近点来构造线性分类边界, 本节在 CDMA 基础上, 考虑两类样本凸可分情况下, 分类边界的构造问题. 首先介绍支持凸线性感知器算法 (SCA), 然后引入极大切割的概念来改进 SCA, 并给出新方法的预测规则和时间复杂度分析.

2.1　支持凸线性感知器算法

基于凸可分 (Convexly separable) 的概念[10], 凸线性感知器 (Convex linear perceptron, Conlitron) 被提出, 它是一组线性函数的集合, 可将任意两类凸可分数据正确分开. 对于 $X, Y \subseteq \mathbf{R}^n$, 如果 X 相对 Y 是凸可分的, 即 $Y \cap CH(X) = \emptyset$, 那么必然存在一个凸线性感知器, 表示如下:

$$CLP = \{f_l(x) = w_l \cdot x + b_l, \\ (w_l, b_l) \in \mathbf{R}^n \times \mathbf{R}, 1 \leq l \leq L\} \quad (4)$$

满足下面两式:

$$\forall\, x \in X, \forall 1 \leq l \leq L,\ \ f_l(x) = w_l \cdot x + b_l > 0$$
$$\forall\, y \in Y, \exists 1 \leq l \leq L,\ \ f_l(y) = w_l \cdot y + b_l < 0$$

此时, 决策函数定义为

$$CLP(x) = \begin{cases} +1, & \forall\, 1 \leq l \leq L,\ \ f_l(x) > 0 \\ -1, & \exists\, 1 \leq l \leq L,\ \ f_l(x) < 0 \end{cases} \quad (5)$$

如果集合 X 相对 Y 是凸可分的, 或者 Y 相对 X 是凸可分的, 则称集合 X 和 Y 是凸可分的.

凸线性感知器的构造通过支持凸线性感知器算法 (SCA) 来实现, 算法 2 给出了在 X 相对 Y 是凸可分的情况下 SCA 算法的描述, SCA 的几何解释见图 2.

算法 2. 支持凸线性感知器算法 (SCA)

Algorithm SCA (X, Y, ε)

1) $l \leftarrow 1$;

2) Repeat

(a) $p = \arg\min_j \{d(CH(X), \{y_j\}), y_j \in Y\}$;

(b) $f_l(x) = \mathrm{CDMA}(X, \{y_p\}, \varepsilon)$;

(c) $Y = \{y | f_l(y) > f_l(y_p), y \in Y\}$;

(d) $l \leftarrow l + 1$;

until $Y = \phi$

3) Return $CLP = \{f_i(x), 1 \leq i \leq l\}$.

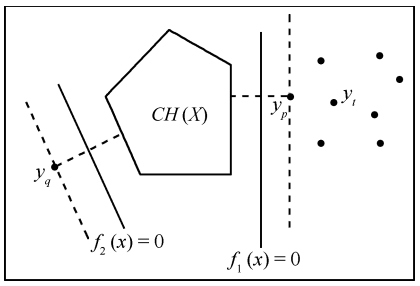


图 2　SCA 算法的几何解释: y_t 满足 $f_1(y_t) \leq f_1(y_p)$, $margin1 = d(CH(X), y_p) = \sqrt{-2f_1(y_p)}$, $margin2 = d(CH(X), y_q) = \sqrt{-2f_2(y_q)}$, 且 $margin1 \leq margin2$

Fig. 2　The geometric explanation of SCA (y_t satisfies $f_1(y_t) \leq f_1(y_p)$, $margin1 = d(CH(X), y_p) = \sqrt{-2f_1(y_p)}$, $margin2 = d(CH(X), y_q) = \sqrt{-2f_2(y_q)}$, and $margin1 \leq margin2$.)

首先, SCA 从集合 Y 中选择一个点 y_p, y_p 是 Y 中距离 $CH(X)$ 最近的点, 利用 CDMA 算法, 计算 y_p 与 $CH(X)$ 的一个线性函数 $f_1(x)$. $f_1(x)$ 切掉了 Y 中满足条件 $f_1(y) \leq f_1(y_p)$ 的点, 剩余的点仍记为 Y. 然后从 Y 中找到距离 $CH(X)$ 另一个最近点 y_q, 计算得到第二个线性函数 $f_2(x)$, 重复这个过程直到 $Y = \emptyset$, 得到的所有线性函数 $\{f_1(x), f_2(x), \cdots\}$ 构成一个凸线性感知器, 它能够将集合 X 和 Y 正确分开.

2.2　极大切割的 SCA

SCA 算法每次从 Y 集合中选择距离 $CH(X)$ 最近的点, 考虑到这样产生的线性函数总数不能够保证尽可能的少, 所以这里我们引入一种极大切割 (Maximal cutting) 的思想来选取尽可能少的线性函数. 首先, 选择能够切掉 Y 中最多点的线性函数 $g_p(x)$ 作为第一个决策函数 (值得注意的是, 在文献 [10] 中选择最近, 而这里寻求最多), 然后去掉 Y 中满足条件 $g_p(y) < 0$ 的点, 剩余的点仍记为 Y. 接下来, 继续选择能够切掉 Y 中最多点的线性函数

$g_q(x)$ 作为第二个决策函数, 重复这个过程, 直到 $Y = \emptyset$. 经过选择, 所有的决策函数构成一个新的凸线性感知器. 由于 Y 是一个有限集, 经过若干次切割后, 这个过程一定会停止, 此时满足 $Y = \emptyset$.

令 $G(x)$ 表示 Y 中每个点与 X 进行训练得到的初始线性函数集, 我们定义一个极大化切割过程 (Maximal cutting process, MCP) 来完成上述极少决策函数的选取. MCP 每次从 $G(x)$ 中选取能够切开 Y 中最多点的线性函数 $g_p(x)$ 作为新的决策函数, 过程描述见算法 3.

为了与后续 SMA 中的相似操作区分, 这里我们取符号 SCA 中的字母 C 来表示 SCA 算法中的这一过程, 即 C-maximal-cutting; 相应地, 后续 SMA 中的过程称为 M-maximal-cutting.

算法 3. 极大切割过程 (C-maximal-cutting)

Process C-maximal-cutting $(Y, G(x))$
1) counter $[i] = 0$, $1 \le i \le |Y|$;
2) for $1 \le i \le |Y|$
$Y_i = \{y | g_i(y) < 0, y \in Y, g_i(x) \in G(x)\}$,
counter $[i] \leftarrow |Y_i|$;
end for
3) $p = \arg\max_i \{\text{counter}[i]\}$;
4) Return p.

每个决策函数 $g_i(x)$ 都将 $CH(X)$ 与一组点 Y_i 分开, 而这个线性函数 $g_i(x)$ 最初却是由 Y 中的单个点和 X 中的所有点经过 CDMA 算法训练得到的, 从统计的观点来看, 由单个点与 $CH(X)$ 构造的这个线性函数并不能代表合理的分类边界. 为了得到更有效的线性边界, 我们使用 Y_i 中的所有点与 X 进行一个二次训练, 使初始得到的线性边界得到调整, 以提高分类器总体的泛化能力. 这个过程称为边界调整 (Boundary adjusting process, BAP), 记作 C-boundary-adjusting, 直观描述见图 3.

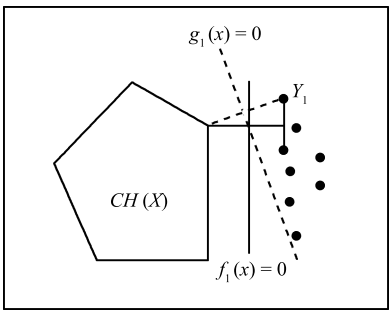


图 3 边界调整的几何解释

Fig. 3 The geometric explanation of C-boundary-adjusting

初始边界 $g_1(x)$ 由单个点 y_1 与 $CH(X)$ 经过 CDMA 算法训练得到, $g_1(x)$ 在图中用虚线表示, 它切开了 $g_1(x)$ 右侧的若干点 Y_1, 经过边界调整后, 由 Y_1 与 $CH(X)$ 共同训练得到了新的边界 $f_1(x)$, 从图 3 中可看出新的线性边界泛化能力更好. 值得注意的是, 分类边界由从 $g_1(x)$ 变为 $f_1(x)$, 即包含着极大切割过程中的线性函数选取, 又包含着边界调整过程中的校正.

极大切割过程和边界调整过程组合在一起, 构成了一种新的 SCA 算法, 称为基于极大切割的 SCA 算法, 记作 MC-SCA.

从算法 4 的描述中, 我们可以清晰地看到, 极大切割过程 MCP 处于该算法的第一阶段, 目的是得到更少的决策函数, 以简化模型结构. 而处于第二阶段的边界调整过程 BAP (即二次训练, 见算法第 3 (d) 步) 是为了得到更为合理的线性边界, 以提高其泛化能力. 两个过程构成了两阶段训练的主体, 尽管它们的构造带有启发性, 但从第 4 节的数值实验中可以看到它们的有效性. 由于集合 X 和 Y 的对称性, 我们很容易得到 Y 相对 X 为凸可分的情况下, 极大切割的 SCA 算法.

算法 4. 极大切割的 SCA 算法 (MC-SCA)

Algorithm MC-SCA (X, Y, ε)
1) $l \leftarrow 1$, $m = |Y|$;
2) $G(x) = \{g_i(x) | g_i(x) = \text{CDMA}(X, \{y_i\}, \varepsilon)$, $1 \le i \le m\}$;
3) Repeat
(a) $p =$ C-maximal-cutting $(Y, G(x))$;
(b) $Y_t = \{y | g_p(y) < 0, y \in Y\}$;
(c) $G_t(x) = \{g_j(x) | y_j \in Y_t\}$;
(d) $f_l(x) = \text{CDMA}(X, Y_t, \varepsilon)$;
(e) $Y = Y - Y_t$;
(f) $G(x) = G(x) - G_t(x)$;
(g) $l \leftarrow l + 1$;
until $Y = \phi$
4) Return $CLP = \{f_i(x), 1 \le i \le l\}$.

2.3 MC-SCA 的预测规则

经过 MC-SCA 训练后, 得到的模型为一组线性函数的集合, 表示为式 (4) 的形式, 即 $CLP = \{f_l(x), 1 \le l \le L\}$. 使用此模型对一个未知样本 z 进行预测时, 要使用式 (5) 的决策规则, 即在 X 相对 Y 为凸可分的情况下, 如果此样本属于 X, 则要满足 $\forall 1 \le l \le L$, $f_l(z) > 0$. 如果此样本属于 Y, 则要满足 $\exists 1 \le l \le L$, $f_l(z) < 0$.

为了缩短预测时间和减小计算代价, 对任意一个未知样本 z, 均从预测它是否属于 Y 开始, 如果存在一个线性函数 $f_l(z) < 0$, 则预测过程立即停止, 并标识 z 为 Y 类. 如果对于整个模型的线性函数集来说, 均满足 $\forall 1 \le l \le L$, $f_l(z) > 0$, 则此时才预测它为 X 类. 这样, 就避免了分别使用 X 类和 Y 类

的规则对未知样本进行预测, 节省了预测时间.

同理可得, 当 Y 相对 X 为凸可分的情况下, MC-SCA 的预测规则.

2.4 MC-SCA 的时间复杂度

无论 X 和 Y 是否凸可分, 由于它们都是有限集, 原始 SCA 算法最终一定能够收敛, 相关证明参见文献 [10] 定理 7. SCA 算法的时间复杂度被估计为 $\mathrm{O}(D(|X|\cdot|Y|)/\varepsilon)$, 与 CDMA 时间复杂度计算方式相同, D 的取值与样本分布和参数 ε 取值相关, 代表了算法收敛的最大迭代次数.

同理 MC-SCA 算法是收敛的, 但其时间复杂度要高于原始 SCA 算法, 为了说明此时间复杂度的构成, 以及方便地同 SCA 进行对比, 我们将其分为两部分来讨论. 第一部分表示在 MC-SCA 不进行边界调整的情况下, 算法所用时间与 SCA 基本相同, 即 $\mathrm{O}(D(|X|\cdot|Y|)/\varepsilon)$. 尽管决策函数的取法不同, SCA 取距离最近, 而 MC-SCA 取切割最多, 但从总体上进行估计, 两者可取得相同的时间复杂度.

第二部分代表了边界调整所用的时间, 在 X 相对 Y 为凸可分的情况下, 每次调整要使用 Y 中的部分点 Y_i 与 X 进行一个 CDMA 训练, 我们考虑最坏的情况, 即每次调整都消耗一个 X 中全部点和 Y 中全部点的 CDMA 训练时间, 即 $\mathrm{O}(D(|X|+|Y|)/\varepsilon)$, 而最多调整次数为 Y 中点的个数 $|Y|$, 由此得到所用时间为 $\mathrm{O}(|Y|\cdot(D(|X|+|Y|)/\varepsilon))$. 同理, 可得到在 Y 相对 X 为凸可分的情况下, 边界调整的时间复杂度为 $\mathrm{O}(|X|\cdot(D(|X|+|Y|)/\varepsilon))$. 最后综合两部分的结果, 在总体上, 将 MC-SCA 算法的时间复杂度大致估计为 $\mathrm{O}(D(|X|\cdot|Y|+(|X|+|Y|)^2)/\varepsilon)$.

值得注意的是, 在实际实现的过程中, 极大切割过程所消耗的时间低于算法中描述的情形, 因为它里面的数组 counter[i] 可以在算法 4 的第 2 步初始得到, 然后在以后的迭代过程中, 根据保留的样本点坐标相应调整即可.

3 极大切割的支持组合凸线性感知器算法

文献 [10] 已经证实, 对于任意不存在重合样本的两类数据, 组合凸线性感知器能够实现完全的分割. 本节在此基础上, 研究基于极大切割的组合凸线性感知器的构造问题, 以实现极少线性函数的选取, 并通过边界调整, 提高感知器的泛化能力.

3.1 支持组合凸线性感知器算法

组合凸线性感知器 (Multiple convex linear perceptron, multiconlitron) 是一组凸线性感知器 (Conlitrons) 的集合. 如果两个有限集 $X, Y \subseteq \mathbf{R}^n$ 是叠可分 (Commonly separable) 的, 即它们没有重合点 ($X \cap Y = \emptyset$), 则一定存在一个组合凸线性感知器 (Multiconlitron) 将其分开. 这个组合凸线性感知器可被表示为 $MCLP = \{CLP_k, 1 \le k \le K\}$, 方向为 X 到 Y, 并满足以下两式:

$$\forall x \in X,\ \exists 1 \le k \le K,\ CLP_k(x) = +1 \tag{6}$$

$$\forall y \in Y,\ \forall 1 \le k \le K,\ CLP_k(y) = -1 \tag{7}$$

组合凸线性感知器的决策函数定义如下:

$$MCLP(x) = \begin{cases} +1,\ \exists 1 \le k \le K,\ CLP_k(x) = +1 \\ -1,\ \forall 1 \le k \le K,\ CLP_k(x) = -1 \end{cases} \tag{8}$$

组合凸线性感知器的构造通过支持组合凸线性感知器算法 (Support multiconlitron algorithm, SMA) 来实现, 算法 5 给出了 SMA 算法的描述, 几何解释见图 4.

算法 5. 支持组合凸线性感知器算法 (SMA)

Algorithm SMA (X, Y, ε)

1) $k \leftarrow 1$;

2) Repeat

(a) $p = \arg\min_i \{d(\{x_i\}, Y), x_i \in X\}$;

(b) $CLP_k = \mathrm{SCA}(\{x_p\}, Y, \varepsilon)$;

(c) $X = \{x_i | \exists f \in CLP_k,\ f(x_i) < f(x_p),\ x_i \in X - \{x_p\}\}$;

(d) $k \leftarrow k + 1$;

until $X = \phi$

3) Return $MCLP = \{CLP_i, 1 \le i \le k\}$.

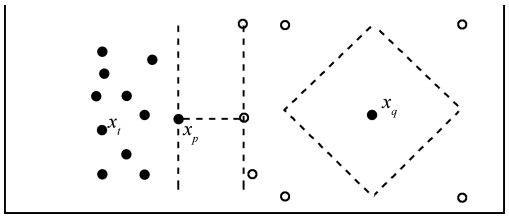


图 4　SMA 算法的几何解释 (x_p 左侧的任意点 x_t 满足 $f_l(x_t) \ge f_l(x_p), \forall f_l \in CLP_1$. 由 x_q 计算得到 CLP_2, CLP_2 中线性函数围成区域中的点 x_q 满足 $CLP_2(x_q) = +1$.)

Fig. 4　The geometric explanation of SMA (For any point x_t from the left side of x_p, it satisfies $f_l(x_t) \ge f_l(x_p)$, $\forall f_l \in CLP_1$. A point x_q surrounded by multiple linear functions in CLP_2 satisfies $CLP_2(x_q) = +1$.)

由于集合 $X \cap Y = \emptyset$, 所以对于 X 中的每一个点 x_i 来说, 它对 Y 都是凸可分的. SMA 算法首先从集合 X 中选择距离 Y 最近的一个点 x_p, 构造第一个凸线性感知器 $CLP_1 = SCA(\{x_p\}, Y, \varepsilon)$, 作为 $MCLP$ 的第一个组件, 它切掉了 X 中满足条件 $f_l(x) \ge f_l(x_p), \forall f_l \in CLP_1$ (即 $CLP_1 = +1$) 的点, 剩余的点仍记为 X. 然后, 再从 X 中找到另

一个最近点 x_q, 得到第二个凸线性感知器 $CLP_2 = SCA(\{x_q\}, Y, \varepsilon)$, 构成$MCLP$的第二个组件. 重复这一进程, 直到 X 中没有点留下来, 即 $X = \emptyset$.

3.2 极大切割的 SMA

考虑到 SMA 算法每次从 X 集合中选择距离 Y 最近 $\min\{d(\{x_i\}, Y), x_i \in X\}$ 的点, 这样产生的线性函数总数不能够保证尽可能的少, 所以这里我们同样引入一个极大切割过程 M-maximal-cutting (见算法 6), 来选取尽可能少的决策函数.

算法 6. 极大切割过程: M-maximal-cutting

Process M-maximal-cutting $(X, GCLP)$
1) counter $[i] = 0$, $1 \le i \le |X|$;
2) for $1 \le i \le |X|$
  $X_i = \{x | gCLP_i(x) = +1, x \in X, gCLP_i \in GCLP\}$;
  counter $[i] \leftarrow |X_i|$;
  end for
3) $p = \arg\max_i \{\text{counter}[i]\}$;
4) Refurn p.

令$GCLP$ 表示 X 中每个点与 Y 进行训练得到的初始凸线性感知器 (以下简称凸线器) 集合, 从$GCLP$ 中选择能够切掉 X 中最多点的 $gCLP_p$ 作为第一个决策凸线器, 然后去掉 X 中满足条件 $gCLP_p(x_i) = +1$ 的点, 剩余的点仍记为 X. 接下来, 继续选择能够切掉 X 中最多点的 $gCLP_q$ 作为第二个决策凸线器, 它又切掉了满足 $gCLP_q(x_i) = +1$ 的这些点, 重复这个过程, 直到 $X = \emptyset$. 这样所有的决策凸线器构成一个组合凸线性感知器$MCLP$. 由于 X 是一个有限集, 经过若干次切割后, 这个过程一定会停止, 即此时满足 $X = \emptyset$.

经过极大切割过程后, 我们能够找到切开 X 中最多点的凸线器 $gCLP_p$, 它分开了一组 X 中的点 X_p 与 Y, 但它只是由 X_p 中的单个点与 Y 训练得到的边界, 从统计的观点看, 这个由 $gCLP_p$ 中所有线性函数围成的边界并不能代表合理的分类面. 所以我们使用 X_p 中的全部样本与 Y 进行一个二次训练, 调整分类边界到合理位置, 以提高分类器的泛化能力, 这个过程我们仍然称为边界调整 (Boundary adjusting process, BAP), 记作 M-boundary-adjusting. 与 C-boundary-adjusting 不同, M-boundary-adjusting 是一组线性边界的调整.

极大切割过程和边界调整过程组合在一起, 构成了新的 SMA 算法, 称为极大切割 (Maximal cutting) 的 SMA 算法, 记作 MC-SMA, 描述如算法 7 所示. 两阶段训练分别能够得到更少的决策凸线器数量和产生更加合理的线性边界, 最终提高泛化能力. 在 MC-SMA 进行凸线器调整的过程中, 包含着 MC-SCA 进行线性边界调整的过程.

算法 7. 极大切割的 SMA 算法 (MC-SMA)

Algorithm MC-SMA (X, Y, ε)
1) $k \leftarrow 1, n = |X|$;
2) $GCLP = \{gCLP_i | gCLP_i = MC - SCA(\{x_i\}, Y, \varepsilon), 1 \le i \le n\}$;
3) Repeat
  (a) p = M-maximal-cutting $(X, GCLP)$;
  (b) $X_t = \{x | gCLP_p(X) = +1, x \in X\}$;
  (c) $GCLP_t = \{gCLP_j | x_j \in X_t\}$;
  (d) CLP_k = MC-SCA(X_t, Y, ε);
  (e) $X = X - X_t$;
  (f) $GCLP = GCLP - GCLP_t$;
  (g) $k \leftarrow k + 1$
  until $Y = \phi$;
4) Return $MCLP = \{CLP_i, 1 \le i \le k\}$.

3.3 MC-SMA 的预测规则

经过 MC-SMA 训练后, 得到的模型为一组凸线器 (Conlitrons) 的集合, 表示为 $MCLP = \{CLP_k, 1 \le k \le K\}$. 使用此模型对一个未知样本 z 进行预测时, 要使用式 (6) 的决策规则, 即在训练方向为 X 到 Y 时, 如果此样本属于 X, 则要满足 $\exists 1 \le k \le K$, $CLP_k(z) = +1$; 如果此样本属于 Y, 则要满足 $\forall 1 \le k \le K$, $CLP_k(z) = -1$.

为了缩短预测时间和减小计算代价, 对任意一个未知样本 z, 均从预测它是否属于 X 开始, 在预测的过程中, 如果存在一个 CLP_k, 使得 $CLP_k(z) = +1$, 则预测过程立即停止, 并标识 z 为 X 类; 如果对于整个模型的凸线器集合来说, 均满足 $\forall 1 \le k \le K$, $CLP_k(z) = -1$, 则此时才预测它为 Y 类.

同理可得, 当训练方向为从 Y 到 X 时, MC-SMA 的预测规则.

3.4 MC-SMA 的时间复杂度

由于 X 和 Y 都是有限集, 原始 SMA 算法也一定能够收敛, 相关证明参见文献 [10] 中的定理 8. 在 SMA 算法中, 分类超平面均由两类间的样本构造, 不涉及凸包间的运算, 所以精度参数 ε 对算法几乎没有影响, 时间复杂度可被评估为 $O(|X| \cdot |Y| \cdot (|X| + |Y|))$.

同理, 可得到 MC-SMA 算法仍然是收敛的, 但其时间复杂度要高于原始 SMA 算法, 为了说明此时间复杂度的构成, 以及能够方便地同 SMA 进行对比, 我们将其分为两部分来讨论. 第一部分表示在 MC-SMA 不进行边界调整的情况下, 算法所用时间与 SMA 基本相同, 即 $O(|X| \cdot |Y| \cdot (|X| + |Y|))$. 尽管两种方法决策凸线器的取法不同, SMA 取距离最近, 而 MC-SMA 取切割最多, 但从总体上来看, 两者可取得相同的时间复杂度.

第二部分代表了边界调整所用的时间, 在训练方向为从 X 到 Y 时, 每次调整要使用 X 中的部分点 X_i 与 Y 进行一个 MC-SCA 训练, 我们考虑最坏的情况, 即每次调整都消耗一个 X 中全部点和 Y 中全部点的 MC-SCA 训练时间, 即 $\mathrm{O}(D(|X|\cdot|Y|+(|X|+|Y|)^2)/\varepsilon)$, 而最多调整次数为 X 中点的个数 $|X|$, 由此得到所用时间为 $\mathrm{O}(|X|\cdot D(|X|\cdot|Y|+(|X|+|Y|)^2)/\varepsilon)$. 同理, 可得到在训练方向为从 Y 到 X 时, 边界调整的时间复杂度为 $\mathrm{O}(|Y|\cdot D(|X|\cdot|Y|+(|X|+|Y|)^2)/\varepsilon)$. 最后综合两部分的结果, 在总体上, 将 MC-SMA 算法的时间复杂度大致估计为 $\mathrm{O}(D((|X|+|Y|)\cdot(|X|^2+|Y|^2))/\varepsilon)$.

与原 SMA 不同, MC-SMA 的边界调整过程包含着样本集凸包间的训练, 所以精度参数 ε 对算法存在一定影响. 值得注意的是, MC-SMA 的边界调整是一个凸线器中所有线性边界的调整, 而 MC-SCA 每次的调整只是一条线性边界的调整.

4　数值实验

本节将通过数值实验来验证 MC-SCA 和 MC-SMA 的性能. 实验分为三部分: 1) 在人工合成数据集上的实验; 2) 本文方法与原 SCA、SMA 算法及线性 SVM (SVM.lin)、RBF 核 SVM (SVM.rbf) 的对比实验; 3) 本文方法与两个典型分片线性分类器 (最近邻 NNA、决策树 DTA) 的对比. 在实验中, 我们使用 *CLP* size 表示一个凸线器中包含的线性函数的数量, *MCLP* size 表示一个组合凸线器中包含的 Conlitrons 数量, LF number 表示一个 Multi-conlitron 包含的线性函数的总数. 所有的实验都在统一的条件下进行, 精度参数设置为 $\varepsilon=10^{-3}$, 处理器 I5-2400 3.10 GHz, 内存 4 GB, Windows 7.0 操作系统.

4.1　人工合成数据集实验

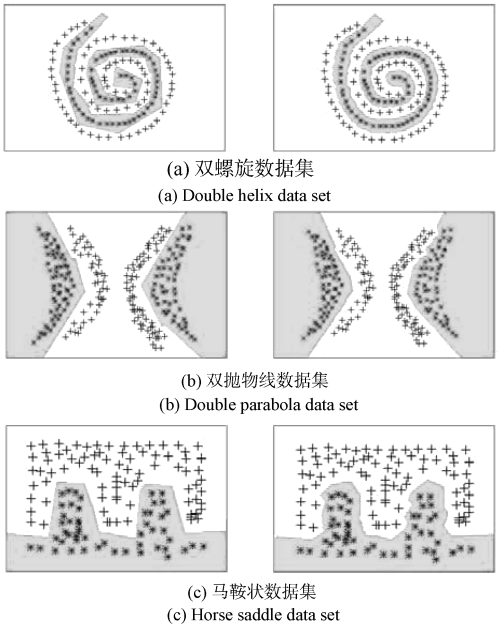
我们首先考虑 3 个人工合成数据集: 双螺旋数据集[12]、双抛物线数据集[12] 和马鞍状数据集[13]. "+" 代表第 1 类样本, "*" 代表第 2 类样本. 对每一个数据集, 分别应用 MC-SMA 和 SMA 算法, 图 5 是在二维平面中的分类效果, 表 1 为相应的凸线器数量和线性函数总数.

作为典型的线性不可分情况, 这 3 个数据集被公认为是模式识别领域的典型问题, 同时也是检验模式识别算法的有效手段. 从图 5 和表 1 中可以看出, MC-SMA 算法得到的线性边界要比 SMA 简单, 线性函数数量更少. 在 3 个数据集上, 线性函数总数都至少减少了 50 % 以上, 在双抛物线数据集上更是达到了 92 %. 通过在以上 3 个人工合成数据集上的对比, 说明 MC-SMA 训练得到的感知器模型要比 SMA 更简单.

表 1　MC-SMA 和 SMA 在人工合成数据集上的对比

Table 1　Comparison of MC-SMA and SMA on three synthetic data sets

数据集名称	MC-SMA		SMA	
	MCLP size	LF number	*MCLP* size	LF number
Double helix	17	42	66	224
Double parabola	6	14	54	185
Horse saddle	5	11	24	90



(a) 双螺旋数据集
(a) Double helix data set

(b) 双抛物线数据集
(b) Double parabola data set

(c) 马鞍状数据集
(c) Horse saddle data set

图 5　3 个人工合成数据集 (每一组左侧使用 MC-SMA, 右侧使用 SMA)

Fig. 5　Experiments on three synthetic data sets (MC-SMA and SMA are applied to the left side and the right side of each group, respectively.)

4.2　标准数据集实验

这一部分以原始算法和 SVMs 为基准, 评估本文方法的分类性能. 为了体现客观性, 我们从 UCI 机器学习数据集[14] 中选择 13 个标准数据集来进行实验, 数据集描述见表 2. 由于最后的 4 个数据集已被 UCI 分成 "训练 + 测试" 两部分, 所以我们直接对其进行训练和测试. 而对于前 9 个数据集中的每一个, 我们分 10 次随机将其切为两半, 一半用作训练, 另一半用作测试, 然后统计它们的平均实验结果.

SVMs 使用两种类型, 带参数 C 的线性 SVM 和带参数 (C,γ) 的 RBF 核 SVM, 参数的值通过 10 折交叉验证来选取, C 和 γ 的候选集分别为 $\{2^i|i=-4,-3,\cdots,3,4\}$ 和 $\{2^i|i=-7,-6,\cdots,4,5\}$. 在

实验中, 我们首先将数据集缩放到 $[0, 1]$ 之间, 然后分别使用 LIBLINEAR[15] 和 LIBSVM[16] 来执行测试.

表 2 13 个 UCI 数据集描述

Table 2 Data sets used in the experiments

名称	缩写	样本数	维度
Breast	BRE	569	30
German	GER	1 000	24
Heart	HEA	297	13
Ionosphere	ION	351	34
Magic	MAG	19 020	10
Musk1	MUS	476	166
Parkinsons	PAR	195	22
Pima	PIM	768	8
Sonar	SON	208	60
Monks-1	MO1	124 + 432	6
Monks-2	MO2	169 + 432	6
Monks-3	MO3	122 + 432	6
Spectf	SPE	80 + 187	44

4.2.1 凸可分实验

表 3 中的三个数据集都是凸可分的, 其凸性判断过程如下, 在精度 $\varepsilon = 10^{-3}$ 的条件下, 利用 SCA 对其进行训练, 算法收敛并得到一组线性函数, 然后利用此线性函数集再对它们进行测试, 正确率为 100 %, 由此我们得到这三个数据集在 ε 精度下是凸可分的, 并且经过切割后依然保持凸性.

在这一部分中, 我们分别使用它们来对 SCA 和 MC-SCA 进行性能评估. 实验结果包括测试精度、线性函数数量 (CLP size)、训练时间和测试时间.

从表 3 中我们可以看出, MC-SCA 的分类精度要比 SCA 略高一些, 而线性函数的数量却有了非常明显的下降. 在 MUS 和 SON 上, 线性函数数量减少了 50 % 以上, 在 SPE 上也减少了超过 30 %, 从凸可分实验上验证了我们改进算法的有效性, 在保证精度的情况下, 使分类模型结构得到简化. 由于引入了两阶段训练过程, 训练时间较 SCA 增加了一些, 但由于决策函数数量减少, 所以测试时间也相应减少.

4.2.2 叠可分实验

表 4 为本文方法 MC-SMA 与原 SMA 及 SVMs (SVM.lin, SVM.rbf) 的对比实验结果, 所用数据集均为叠可分、类间无重合样本. 性能评价指标包括测试精度、训练时间和测试时间, 另外, 为了说明引入极大切割过程后, 生成模型的简化程度, 表 5 给出了 MC-SMA 与 SMA 在凸线器数量 ($MCLP$ size)、线性函数总数 (LF number) 上的对比.

从表 4 中我们可以看出, MC-SMA 在 9 个数据集 (GER, HEA, ION, MAG, MUS, MO1, MO2, MO3, SPE) 上分类精度要高于 SMA, 并且在其中的一些数据集上, 精度提高明显, 如在 MO1 和 MO3 上分别提高了 11.67 % 和 10.88 %. 在另外的 4 个数据集 (BRE, PAR, PIM, SON) 上分类精度略低于 SMA, 采用 T-test 检验后发现, 除 PIM 外, MC-SMA 在 BRE、PAR、SON 这 3 个数据集上获得的分类精度与 SMA 并无显著差别. 所以总体上 MC-SMA 的分类精度要好于 SMA. 由于引入了边界调整进程 BAP, MC-SMA 的训练时间要多于 SMA.

与 SVMs 的对比实验中, MC-SMA 在 10 个数据集 (除 GER、HEA 和 SPE 外) 上分类精度要高于线性 SVM, 最大的正差值为 28.70 % (MO1), 最大的负差值为 7.33 % (HEA); 但只在 3 个数据集 (PAR, MO1, MO2) 上好于 RBF 核 SVM, 提高 MC-SMA 的性能以接近或达到核 SVM 的水平, 仍然是今后工作努力的方向. 另外, 如表 4 所示, 在不考虑参数寻优的情况下, SVMs 所消耗的训练时间要少于 MC-SMA.

表 5 对比了本文方法与原 SMA 在凸线器数量 ($MCLP$ size) 和线性函数总数 (LF number) 上的实验结果. 在所有的数据集上, 由 MC-SMA 得到的线性函数数量均小于 SMA, 且都至少减少了 50 % 以上. 减少最多的为 MUS, SMA 得到线性函数总数为 5 095, 而 MC-SMA 为 126, 减少了大约 97.5 %, 在 ION、SON 和 SPE 上, 都减少了 90 % 以上. 从标准数据集的实验中证明, MC-SMA 确实能够得到更简化的模型.

4.3 与 NNA 和 DTA 的对比实验

最近邻分类器 (Nearest neighbor classifier, NNA) 是广为熟知的机器学习算法[17], 一个未知样

表 3 SCA 与 MC-SCA 的对比

Table 3 Comparison of SCA and MC-SCA

数据集名称	测试精度 (%)		CLP size		训练时间 (秒)/测试时间 (毫秒)	
	SCA	MC-SCA	SCA	MC-SCA	SCA	MC-SCA
MUS	85.48	85.56	120	46	6.806/4.40	9.717/3.57
SON	78.67	79.33	45	18	0.021/0.35	0.033/0.18
SPE	59.36	63.10	33	22	0.047/0.30	0.078/0.16

表 4 本文方法 MC-SMA 与 SMA、SVMs 的对比

Table 4 Comparison of MC-SMA and SMA, SVMs

数据集名称	测试精度 (%)				训练时间 (秒)/测试时间 (毫秒)			
	MC-SMA	SMA	SVM.lin	SVM.rbf	**MC-SMA**	SMA	SVM.lin	SVM.rbf
BRE	91.79	91.86	90.60	93.33	1.340/0.67	0.043/0.90	0.001/0.06	0.017/2.06
GER	67.68	64.28	72.18	74.00	4.625/6.61	0.333/9.30	0.003/0.09	0.034/14.08
HEA	62.30	59.41	69.63	71.48	0.179/0.44	0.019/0.59	0.001/0.02	0.013/0.98
ION	88.35	86.93	84.77	92.90	0.264/0.20	0.052/0.94	0.002/0.04	0.020/1.90
MAG	79.19	77.45	78.81	82.97	7 089.000/504.33	167.630/3422.00	0.018/1.15	3.811/2262.23
MUS	83.81	82.97	80.79	89.96	11.725/3.34	0.729/11.90	0.011/0.16	0.058/15.00
PAR	82.14	82.76	78.06	81.22	0.064/0.13	0.008/0.25	0.002/0.02	0.010/0.43
PIM	66.22	67.89	65.76	75.47	2.212/1.52	0.117/4.32	<0.001/0.04	0.017/6.21
SON	78.38	79.71	73.81	82.67	0.122/0.71	0.037/1.10	0.005/0.03	0.018/1.21
MO1	97.22	85.65	68.52	86.34	0.047/0.16	0.006/0.30	<0.001/0.05	0.005/1.58
MO2	82.41	81.02	61.34	79.63	0.094/0.75	0.017/1.00	<0.001/0.21	0.008/2.45
MO3	94.44	83.56	73.15	95.14	0.047/0.31	0.006/1.00	<0.001/0.05	0.005/1.07
SPE	62.03	60.43	62.03	70.05	0.109/0.60	0.019/1.00	<0.001/0.06	0.009/1.76

表 5 MC-SMA 与 SMA 在凸线器数量和线性函数总数上的对比

Table 5 Comparison of MC-SMA and SMA on *MCLP* size and LF number

算法	BRE	GER	HEA	ION	MAG	MUS	PAR	PIM	SON	MO1	MO2	MO3	SPE
MCLP size: MC-SMA	16	99	29	19	1 322	45	10	69	19	11	33	12	22
MCLP size: SMA	28	147	57	56	2 961	134	18	130	48	18	57	22	37
LF number: MC-SMA	42	459	104	30	9 687	126	29	299	72	44	165	45	53
LF number: SMA	96	1 923	401	462	65 567	5 095	70	1 112	1 027	146	344	117	861

表 6 MC-SMA 同 NNA、DTA 的精度对比

Table 6 Accuracy comparison of MC-SMA and NNA, DTA

算法	BRE	GER	HEA	ION	MAG	MUS	PAR	PIM	SON	MO1	MO2	MO3	SPE
NNA	91.89	65.04	60.59	84.32	77.27	82.72	83.06	67.97	79.33	83.80	79.40	84.95	60.43
DTA	89.05	68.18	63.41	86.53	67.94	69.62	73.78	69.17	71.24	65.74	60.42	85.65	66.84
MC-SMA	91.79	67.68	62.30	88.35	79.19	83.81	82.14	66.22	78.38	97.22	82.41	94.44	62.03
−NNA	−0.10	**2.64**	**1.71**	**4.03**	**1.92**	**1.09**	−0.92	−1.75	−0.95	**13.42**	**3.01**	**9.49**	**1.60**
−DTA	**2.74**	−0.50	−1.11	**1.82**	**11.25**	**14.19**	**8.36**	−2.95	**7.14**	**31.48**	**21.99**	**8.79**	−4.81

本根据它与其他样本的近邻关系进行分类, 在文献 [18] 中说明了最近邻分类器在本质上是一种分片线性分类器, 在这一部分中将对比 MC-SMA 与 NNA 的分类精度. 另外, 我们也对比由 Kostin 提出的一种决策树分析 (Decision tree analysis, DTA) 方法[9], 它通过不断迭代产生划分子树, 然后根据子树所代表的两类样本质心来构造分片线性分类边界.

为了更加直观地进行对比, 表 6 中最后两行给出了 MC-SMA 与 NNA、DTA 的精度差值. 从表 6 中可以看出, MC-SMA 在 9 个数据集上, 测试精度要高于 NNA, 在另外的 4 个数据集上, 略低于 NNA. 在 MO1 上, 取得最大的正差值 13.42, 在 PIM 上取得最大的负差值 −1.75. 在与 DTA 的对比实验中, 9 个数据集测试精度要高于 DTA. 在 MO1 取得最大的正差值 31.48, 在 PIM 上取得最大的负差值 −2.95. 本文方法分别在 4 个数据集上, 分类精度低于 NNA 和 DTA. 我们对这些数据集上的实验结果进行了 T-test 检验, 除 PIM 外, 在另外的数据集上, MC-SMA 所获得的分类精度与 NNA、DTA 并无显著差别.

通过与 NNA 和 DTA 这些传统方法相对比, 说明 MC-SMA 具有明显的竞争力.

5 总结

本文提出了一种组合凸线性感知器的极大切割构造方法, 包括 MC-SCA 和 MC-SMA. 该方法由两阶段训练来完成, 第一阶段使用极大切割过程 MCP 来构造极小的决策函数集, 最大程度减少决策函数的数量, 最终简化分类模型. 第二阶段使用边界调整过程 BAP 对初始得到的边界进行二次训练, 使其调整到更合理的位置, 提高分类器的泛化能力.

在人工和标准数据集上的实验说明, 本文方法的性能较 SMA 有明显提高, 这可根据奥卡姆剃刀 (Occam′s razor) 准则来解释, 该准则旨在优先选择拟合数据的最简单的假设. 本文方法 MC-SMA 在满足对训练集正确划分的条件下, 包含的线性函数更少, 产生的预测模型更简单, 所以具有更好的分类能力.

同时应看到, 该方法提高了分类精度, 但训练时间有所增加, 主要原因是经过极大切割后需要进行边界调整, 在今后的工作要着重加强此过程的优化和提高. 在与 SVMs 的对比中, 证实了本文方法分类精度一般优于线性 SVM, 但要低于 RBF 核 SVM. 因此, 在今后的工作中可以考虑引入新的方法来进一步提升 MC-SMA 的分类性能, 如将聚类与凸包快速计算相结合[19−20], 实现凸线性感知器的快速构造; 如利用 AdaBoost 将若干弱分类器提升为分类精度高的强分类器的方法[21], 来实现组合凸线性感知器的有效集成等.

References

1 Webb D. Efficient Piecewise Linear Classifiers and Applications [Ph. D. dissertation], University of Ballarat, Australia, 2010

2 Nilsson N J. *Learning Machines*. New York: McGraw-Hill, 1965. 1−137

3 Mangasarian O L. Multisurface method of pattern separation. *IEEE Transactions on Information Theory*, 1968, **14**(6): 801−807

4 Herman G T, Yeung K T D. On piecewise-linear classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992, **14**(7): 782−786

5 Sklansky J, Michelotti L. Locally trained piecewise linear classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1980, **2**(2): 101−111

6 Park Y, Sklansky J. Automated design of multiple-class piecewise linear classifers. *Journal of Classification*, 1989, **6**(1): 195−222

7 Tenmoto H, Kudo M, Shimbo M. Piecewise linear classifiers with an appropriate number of hyperplanes. *Pattern Recognition*, 1998, **31**(11): 1627−1634

8 Cai B B, Huang T, Zhuang X H, Zhao Y X, Sklansky J. Piecewise linear classifiers using binary tree structure and genetic algorithm. *Pattern Recognition*, 1996, **29**(11): 1905−1917

9 Kostin A. A simple and fast multi-class piecewise linear pattern classifier. *Pattern Recognition*, 2006, **39**(11): 1949−1962

10 Li Y J, Liu B, Yang X W, Fu Y Z, Li H J. Multiconlitron: a general piecewise linear classifier. *IEEE Transactions on Neural Networks*, 2011, **22**(2): 276−289

11 Keerthi S S, Shevade S K, Bhattacharyya C, Murthy K R K. A fast iterative nearest point algorithm for support vector machine classifier design. *IEEE Transactions on Neural Networks*, 2000, **11**(1): 124−136

12 Abdallah F, Richard C, Lengelle R. An improved training algorithm for nonlinear kernel discriminants. *IEEE Transactions on Signal Processing*, 2004, **52**(10): 2798−2806

13 Gai K, Zhang C S. Learning discriminative piecewise linear models with boundary points. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence. Georgia, USA: AAAI, 2010. 444−450

14 Frank A, Asuncion A. UCI machine learning repository [Online], available: http://archive.ics.uci.edu/ml, April 20, 2012

15 Fan R E, Chang K W, Hsieh C J, Wang X R, Lin C J. LIBLINEAR: a library for large linear classification. *Journal of Machine Learning Research*, 2008, **9**: 1871−1874

16 Chang C C, Lin C J. Libsvm: a library for support vector machines [Online], available: http://www.csie.ntu.edu.tw/cjlin/libsvm, June 25, 2012

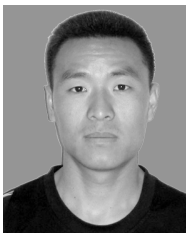
17 Cover T, Hart P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 1967, **13**(1): 21−27

18 Fukunage K. Statistical pattern recognition. *Handbook of Pattern Recognition and Computer Vision (4th edition)*. New Jersey: World Scientific, 2010. 33−60

19 Zhou Lin, Ping Xi-Jian, Xu Sen, Zhang Tao. Cluster ensemble based on spectral clustering. *Acta Automatica Sinica*, 2012, **38**(8): 1335−1342
(周林, 平西建, 徐森, 张涛. 基于谱聚类的聚类集成算法. 自动化学报, 2012, **38**(8): 1335−1342)

20 Liu Bin, Wang Tao. An efficient convex hull algorithm for planar point set based on recursive method. *Acta Automatica Sinica*, 2012, **38**(8): 1375−1379
(刘斌, 王涛. 一种高效的平面点集凸包递归算法. 自动化学报, 2012, **38**(8): 1375−1379)

21 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)



冷强奎 北京工业大学计算机学院博士研究生. 主要研究方向为模式识别与机器学习. E-mail: qkleng@gmail.com
(**LENG Qiang-Kui** Ph. D. candidate at the College of Computer Science, Beijing University of Technology. His research interest covers pattern recognition and machine learning.)

李玉鑑 北京工业大学计算机学院教授. 主要研究方向为模式识别与机器学习. 本文通信作者.
E-mail: liyujian@bjut.edu.cn
(**LI Yu-Jian** Professor at the College of Computer Science, Beijing University of Technology. His research interest covers pattern recognition and machine learning. Corresponding author of this paper.)