第3章

基于加权堆叠选择集成的传统多标签学习

在多标签学习中，多标签分类是重点应用领域，如疾病诊疗、图像标注、基因预测等。现有大量的传统多标签集成方法是基于 bagging 和 boosting 的集成模式，如 EBR、ELP、ECC、EPS、RAkEL、CDE、RF-PCT、AdaBoost. MH 等，基于 stacking 的集成算法较少，代表的有 MLS。然而，这些集成方法在实际应用中存在两个局限：一是在集成策略上主要依赖投票或加权投票的方式，这种方法较为直接但缺乏灵活性，尤其是对于分类器的加权选择策略考虑不足，未能充分利用各分类器间的性能差异来优化集成效果；二是没有考虑标签之间成对的依赖关系，如共生、互斥。为了解决这些问题，本章提出了一种基于 stacking 模式的加权堆叠选择集成算法 MLWSE。该算法可以使用任意的多标签方法作为基学习器，其具有较强的扩展性，而且大量的实验表明了该算法在多标签分类任务中取得了显著的效果提升。

3.1 引言

集成学习算法通过将来自异质或同质模型的单个学习器结合起来，获得一个集成的学习器，能有效处理模型过拟合，提高模型的学习泛化性，广泛应用于多个领域。近年来，许多集成方法被用作多标签分类任务的基准[94]，它们通常采用 bagging 方案生成不同的分类器作为集成成员，并通过多数投票策略获得最后的集成结果。在测试阶段，每个类的预测结果是通过平均每个分类器的置信度来确定的，而没有考虑标签之间分类器选择权重，忽略了局部成对标签依赖关系的影响。尽管堆叠集成方法 stacking 在许多学习任务中具有出色的性能，但也忽视了局部标签之间的依赖关系。MLS[17]可以视为堆叠集成技术的代表，它首先为每个标签

训练独立的二值分类器(一级),然后将它们的预测作为元级学习模型的输入,最后利用共识函数(元级分类器)对多个标签进行堆叠集成,得到最终的预测结果。虽然 MLS 在元级层上考虑了标签间的全局相关性,但仍然忽略了局部成对标签依赖关系的影响,此外,现有的堆叠集成方法也没有考虑分类器的选择权重。

为了解决上述问题,我们同时利用加权堆叠集成和成对标签依赖关系的优点,提出了一种加权分类器选择和堆叠集成的多标签分类算法 MLWSE。在 MLWSE 中,对于不同的类标签,给每个基分类器赋予不同的权重,即任何两个强相关的类标签都比两个不相关或弱相关的类标签具有较高的相似权重。与现有的堆叠集成方法不同,MLWSE 不仅利用稀疏正则实现了分类器选择和集成,而且学习到了标签元级的特别特征。本章可以概括如下:

(1) 本章介绍一种基于 stacking 的多标签分类加权堆叠选择集成算法 MLWSE,该结构采用稀疏正则化方法进行分类器选择和堆叠集成,并可以使用任意的多标签方法作为基学习器,该算法具有较强的扩展性。

(2) MLWSE 算法同时利用了分类器权重和成对标签关联来选择标签元级特别特征,可以当作一种标签元级特别特征选择方法。

(3) MLWSE 算法在二维仿真数据、13 个 Benchmark 基准数据集和真实的心脑血管疾病数据集上进行了实验验证,MLWSE 算法具有较强的鲁棒性和有效性。

3.2　问题描述

在多标签分类中,令$\mathcal{X}=\mathbb{R}^d$表示 d 维的输入空间,$\mathcal{Y}=\{y_1,y_2,\cdots,y_l\}$表示有 l 个类的类标空间,$\mathcal{D}=\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\mid 1\leqslant i\leqslant n\}$表示有 n 个实例的训练集。对每个多标签样本$(\boldsymbol{x}_i,\boldsymbol{y}_i)$,$\boldsymbol{x}_i=[x_{i1},x_{i2},\cdots,x_{id}]\in\mathcal{X}$表示 d 维的特征向量,$\boldsymbol{y}_i=[y_{i1},y_{i2},\cdots,y_{id}]$表示 $\boldsymbol{x}_i$ 的真实类标,当标签 y_j 属于 $\boldsymbol{x}_i$ 时,$y_{ij}=1$;否则,$y_{ij}=0$。多标签学习的任务是从训练集$\mathcal{D}$中学习一个映射关系 $h:\mathcal{X}\to 2^{\mathcal{Y}}$。在测试阶段,对看不见的样本 $\boldsymbol{x}\in\mathcal{X}$,多标签分类器 h 的预测 $h(\boldsymbol{x})\subseteq\mathcal{Y}$可以当作样本 $\boldsymbol{x}$ 的近似。标记输入数据作为矩阵 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]^{\mathrm{T}}\in\mathbb{R}^{n\times d}$,输出作为标签矩阵 $\boldsymbol{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_n]^{\mathrm{T}}\in\mathbb{R}^{n\times l}$。

对于 $n\times d$ 的矩阵$\boldsymbol{A}=[A_{i,j}]$,其中 $i\in\{1,2,\cdots,n\}$,$j\in\{1,2,\cdots,d\}$;用$\boldsymbol{A}^{\mathrm{T}}$表示 $\boldsymbol{A}$ 的转置矩阵;$\mathrm{tr}(\boldsymbol{A})=\sum_{i=1}^{n}A_{i,i}$ 表示 $\boldsymbol{A}$ 的迹;$\|\boldsymbol{A}\|_{\mathrm{F}}=\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{d}A_{i,j}^2}$ 表示 Frobenius 范数;对于任意一个向量$\boldsymbol{a}=[a_1,a_2,\cdots,a_n]^{\mathrm{T}}$,$l_2$-norm 表示为$\|\boldsymbol{a}\|_2=\sqrt{\sum_{i=1}^{n}a_i^2}$,$l_1$-norm 表示为$\|\boldsymbol{a}\|_1=\sum_{i=1}^{n}|a_i|$。

如图 3.1 所示，加权的堆叠集成最小化加权的预测得分 $\boldsymbol{Sw}$ 和真实的目标向量 $\boldsymbol{y}$ 之间的欧几里得距离可描述为

$$\min_{\boldsymbol{w}} \| \boldsymbol{y} - \boldsymbol{Sw} \|_2^2 \tag{3.1}$$

式中：$\boldsymbol{S}$ 为预测的得分矩阵，$\boldsymbol{w}$ 为加权向量，$\boldsymbol{y}$ 为给定数据点的真实目标向量。

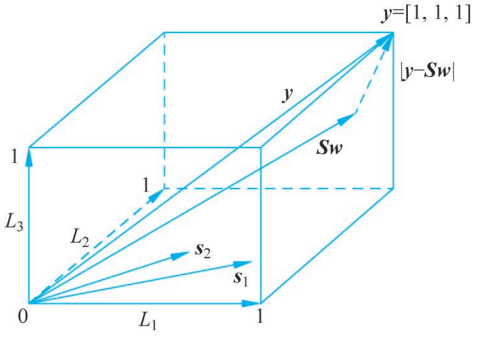


图 3.1 加权选择集成的三维超立方体（目的是最小化预测得分 $\boldsymbol{Sw}$ 和目标向量 $\boldsymbol{y}$ 之间的欧几里得距离）

3.3 MLWSE 算法设计

根据式(3.1)提出的 MLWSE 算法主要分为四个步骤：一是加权的堆叠集成；二是基于稀疏正则化的分类器选择；三是标签依赖关系的建模；四是多标签的预测。

3.3.1 加权的堆叠集成

在具有置信度输出的分类器集成问题中，集成过程是将基分类器获得的属于不同标签的预测分数作为元级分类器的输入。令 s_j^k 表示第 k 个分类器对第 j 个标签的预测得分，则 $\boldsymbol{s}^k = [s_1^k, s_2^k, \cdots, s_l^k]^{\mathrm{T}}$ 表示分类器 k 为所有标签的预测得分，那么集成所有基分类器输入表示为 $s = [\boldsymbol{s}^1 | \boldsymbol{s}^2 | \cdots | \boldsymbol{s}^m]$，其中 m 表示分类器个数，则最后的置信度得分矩阵 $\boldsymbol{S} = [s_{ij}^k]$ 表示为

$$\boldsymbol{S} = \begin{bmatrix} \overbrace{s_{11}^1 & s_{12}^1 & \cdots & s_{1l}^1}^{\boldsymbol{s}^1} \cdots \overbrace{s_{11}^k & s_{12}^k & \cdots & s_{1l}^k}^{\boldsymbol{s}^k} \cdots \\ s_{21}^1 & s_{22}^1 & \cdots & s_{2l}^1 \cdots s_{21}^k & s_{22}^k & \cdots & s_{2l}^k \cdots \\ \vdots & \vdots & \ddots & \vdots \ddots \vdots & \vdots & \ddots & \vdots \cdots \\ s_{n1}^1 & s_{n2}^1 & \cdots & s_{nl}^1 \cdots s_{n1}^k & s_{n2}^k & \cdots & s_{nl}^k \cdots \end{bmatrix}$$

在 stacking 集成模式中，元级集成被定为一种映射 $g:\mathbb{R}^{m\times l}\to\mathbb{R}^{l}$，也就是说，元级分类器最终的目的是使用新产生的数据集 $\{\{(\boldsymbol{s}^{i},y_{ij})\}_{j=1}^{l}\}_{i=1}^{n}$ 学习函数 g，结合式(3.1)，目标函数被最小化为

$$g(\boldsymbol{w}^{1},\boldsymbol{w}^{2},\cdots,\boldsymbol{w}^{m})=\sum_{i=1}^{n}\sum_{j=1}^{l}\Big(\sum_{k=1}^{m}(s_{ij}^{k}w_{j}^{k}-y_{ij})\Big)^{2} \tag{3.2}$$

式中：w_{j}^{k} 表示分类器 k 为标签 j 的权重，且 $\boldsymbol{w}^{k}=[w_{1}^{k},w_{2}^{k},\cdots,w_{l}^{k}]$ 是分类器 k 的权重向量。令 $\boldsymbol{W}_{j}=[\boldsymbol{w}_{j}^{1}|\boldsymbol{w}_{j}^{2}|\cdots|\boldsymbol{w}_{j}^{m}]^{\mathrm{T}}$ 表示所有分类器为 j-th 个标签的联合权重向量，$\boldsymbol{Y}_{j}=[y_{1j},y_{2j},\cdots,y_{nj}]^{\mathrm{T}}$ 表示在标签空间 $\boldsymbol{Y}$ 中标签的第 j 列($1\leqslant j\leqslant l$)，基于产生的置信度的得分矩阵，式(3.2)可进一步表示为

$$\min_{\boldsymbol{W}_{j}}\frac{1}{2}\|\boldsymbol{S}\boldsymbol{W}_{j}-\boldsymbol{Y}_{j}\|_{2}^{2} \tag{3.3}$$

3.3.2　基于稀疏正则的分类器选择

在式(3.3)中，产生的置信度得分矩阵 $\boldsymbol{S}$ 也许包含对标签无帮助且不相关的预测信息，因此不同的分类器对不同的标签应该分配不同的权重。为实现分类器的选择，通过增加稀疏正则保证权重稀疏，以阻止堆叠集成联合所有的分类器。使用稀疏正则的一个好处是它能自动完成选择，因此通过使用 l_{1}-norm 正则(Lasso)[95] 为每个权重向量 $\boldsymbol{W}_{j}$，式(3.3)可进一步描述为

$$\min_{\boldsymbol{W}_{j}}\frac{1}{2}\|\boldsymbol{S}\boldsymbol{W}_{j}-\boldsymbol{Y}_{j}\|_{2}^{2}+\alpha\|\boldsymbol{W}_{j}\|_{1} \tag{3.4}$$

式中：α 为正则参数。通过把所有的二值分类器联合，式(3.4)可写为

$$\min_{\boldsymbol{W}}\frac{1}{2}\|\boldsymbol{S}\boldsymbol{W}-\boldsymbol{Y}\|_{F}^{2}+\alpha\|\boldsymbol{W}\|_{1} \tag{3.5}$$

如果 $w_{j}^{k}=0$，那么表示 k-th 个分类器被排除且没有影响对 j-th 个标签。在 l_{1}-norm 中不是所有的 w_{j}^{k} 都是零，也就意味着所选分类器对某些标签的信息不能被有效利用。如图 3.2 所示，组稀疏 Lasso 不仅考虑了分类器之间的稀疏，而且考

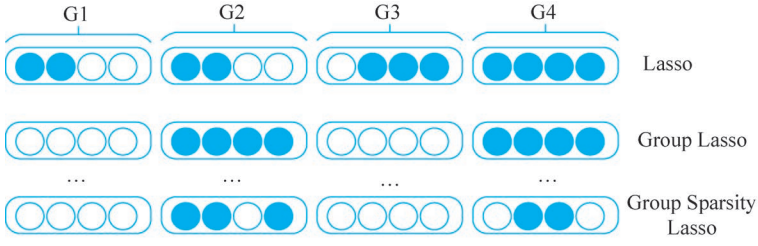


图 3.2　Lasso、Group Lasso 和 Group Sparsity Lasso 比较

虑了分类器内部之间的稀疏，综合了 Lasso 和 Group Lasso 的优点，最终基于 Group Sparsity Lasso，MLWSE 可以表示为

$$\min_{\boldsymbol{W}} \frac{1}{2} \| \boldsymbol{SW}-\boldsymbol{Y} \|_F^2 + \alpha\lambda \| \boldsymbol{W} \|_1 + (1-\alpha)\lambda \sum_{k=1}^{m} c_k \| \boldsymbol{W}_{G_k} \|_2 \tag{3.6}$$

式中：$\alpha \in [0,1]$，用于正则 Lasso 和 Group Lasso；λ 控制了稀疏度；c_k 为 k-th 组 $\boldsymbol{W}_{G_k}$ 的权重，是一种先验为 k-th 分类器选择的贡献，实验中设置 $c_k = \sqrt{l}$。

3.3.3 标签依赖关系的建模

在多标签分类任务中，标签依赖关系是至关重要的，正如多任务学习[97]，任务和模态之间存在着依赖关系。若标签 y_j 和 y_k 是强相关的，则分类器分类标签 y_j 有高的概率分类 y_k。换句话说，如果两个标签 y_j 和 y_k 是强相关的，那么权重向量对 $(\boldsymbol{W}_j, \boldsymbol{W}_k)$ 应该有高的相似；否则，应该有低的相似。通过在标签空间重建一个图 $<V,E>$，V 表示标签的集合，E 表示每对标签之间的边集合，给定标签相关矩阵 $\boldsymbol{R}$ 在 E，建模标签的依赖关系能被最小化为下式：

$$\frac{1}{2} \sum_{j=1}^{l} \sum_{k=1}^{l} \| \boldsymbol{W}_j - \boldsymbol{W}_k \|^2 R_{jk} = \mathrm{tr}(\boldsymbol{W}(\boldsymbol{D}-\boldsymbol{R})\boldsymbol{W}^{\mathrm{T}}) = \mathrm{tr}(\boldsymbol{WHW}^{\mathrm{T}}) \tag{3.7}$$

式中：$\boldsymbol{H} = \boldsymbol{D} - \boldsymbol{R}$ 为图的拉普拉斯矩阵，$\boldsymbol{D}$ 为对角矩阵，$D_{ii} = \sum_{j=1}^{n} R_{ij}$；$R_{jk}$ 为标签 y_j 和 y_k 之间的相似度，本书使用余弦相似度计算标签相关性矩阵。

联立式(3.5)和式(3.7)，基于 Lasso 的 MLWSE-L1 可表示为

$$\min_{\boldsymbol{W}} \frac{1}{2} \| \boldsymbol{SW}-\boldsymbol{Y} \|_F^2 + \alpha \| \boldsymbol{W} \|_1 + \frac{\beta}{2} \mathrm{tr}(\boldsymbol{WHW}^{\mathrm{T}}) \tag{3.8}$$

联立式(3.6)和式(3.7)，基于 Group Sparsity Lasso 的 MLWSE-L21 可表示为

$$\min_{\boldsymbol{W}} \frac{1}{2} \| \boldsymbol{SW}-\boldsymbol{Y} \|_F^2 + \alpha\lambda \| \boldsymbol{W} \|_1 + (1-\alpha)\lambda \sum_{k=1}^{m} c_k \| \boldsymbol{W}_{G_k} \|_2 + \frac{\beta}{2} \mathrm{tr}(\boldsymbol{WHW}^{\mathrm{T}}) \tag{3.9}$$

3.3.4 多标签的预测

使用算法 MLWSE-L1 和 MLWSE-L21 之后，可得到分类器的权重矩阵 $\boldsymbol{W}^*$。当给定使用特征矩阵 $\boldsymbol{X}^*$ 表示的测试数据集时，使用不同的基分类器生成置信度得分矩阵 $\boldsymbol{S}^*$，则可以使用阈值符号函数 $\mathrm{sign}: \mathcal{X} \to \mathbb{R}$ 获得最后的预测结果(实验中阈值 τ 设置为 0.5)：

$$\mathrm{sign}(\boldsymbol{S}^*\boldsymbol{W}^*, \tau) = \begin{cases} 1, & \boldsymbol{S}^*\boldsymbol{W}^* \geqslant \tau \\ 0, & \text{其他} \end{cases} \tag{3.10}$$

3.4　MLWSE 算法优化

尽管式(3.8)和式(3.9)是两个凸的优化问题，由于使用 l_1-norm 正则化，目标函数是非平滑的，本节使用加速的近端梯度下降[98-99]和块坐标下降[100]算法来优化 MLWSE-L1 和 MLWSE-L21。

3.4.1　MLWSE-L1 优化

通常情况下，加速的近端梯度可描述为下面的凸优化问题[99]：

$$\min_{\boldsymbol{W}\in\mathbb{H}}\{F(\boldsymbol{W})=f(\boldsymbol{W})+g(\boldsymbol{W})\} \tag{3.11}$$

式中：$\mathbb{H}$ 是希尔伯特(Hilbert)空间；$f(\boldsymbol{W})$是凸的并且平滑的；$g(\boldsymbol{W})$是凸的，可以是非平滑的。如果 $f(\boldsymbol{W})$有一个利普希茨(Lipschitz)连续梯度，通过使用 Lipschitz 常数 L，则有

$$\| \nabla f(\boldsymbol{W}_1)-\nabla f(\boldsymbol{W}_2) \| \leqslant L \| \boldsymbol{W}_1-\boldsymbol{W}_2 \|$$

代替直接的最小化 $F(\boldsymbol{W})$，近端梯度算法可以最小化其复合二次逼近：

$$Q_L(\boldsymbol{W},\boldsymbol{W}^{(t)})=f(\boldsymbol{W}^{(t)})+\langle\nabla f(\boldsymbol{W}^{(t)}),\boldsymbol{W}-\boldsymbol{W}^{(t)}\rangle+\frac{L}{2}\|\boldsymbol{W}-\boldsymbol{W}^{(t)}\|_F^2+g(\boldsymbol{W}) \tag{3.12}$$

根据式(3.8)和式(3.11)，$f(\boldsymbol{W})$和 $g(\boldsymbol{W})$可写为

$$f(\boldsymbol{W})=\frac{1}{2}\|\boldsymbol{S}\boldsymbol{W}-\boldsymbol{Y}\|_F^2+\frac{\beta}{2}\mathrm{tr}(\boldsymbol{W}\boldsymbol{H}\boldsymbol{W}^{\mathrm{T}}) \tag{3.13}$$

$$g(\boldsymbol{W})=\alpha\|\boldsymbol{W}\|_1 \tag{3.14}$$

根据式(3.13)，可得

$$\nabla f(\boldsymbol{W})=\boldsymbol{S}^{\mathrm{T}}(\boldsymbol{S}\boldsymbol{W}-\boldsymbol{Y})+\beta\boldsymbol{W}\boldsymbol{H} \tag{3.15}$$

给定 $\boldsymbol{W}_1$ 和 $\boldsymbol{W}_2$，为 MLWSE-L1，可获得 Lipschitz 常数[101-102]：

$$L=\sqrt{2\|\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S}\|_2^2+2\|\beta\boldsymbol{H}\|_2^2} \tag{3.16}$$

根据式(3.12)、式(3.14)和式(3.16)，令

$$\boldsymbol{Z}^{(t)}=\boldsymbol{W}^{(t)}-\frac{1}{L}\nabla f(\boldsymbol{W}^{(t)})$$

权重矩阵 $\boldsymbol{W}$ 可优化为

$$\begin{aligned}\boldsymbol{W}^*&=\mathrm{argmin}_{\boldsymbol{W}}Q_L(\boldsymbol{W},\boldsymbol{W}^{(t)})\\&=\mathrm{argmin}_{\boldsymbol{W}}\frac{1}{2}\|\boldsymbol{W}-\boldsymbol{Z}^{(t)}\|_F^2+g(\boldsymbol{W})\\&=\mathrm{argmin}_{\boldsymbol{W}}\frac{1}{2}\|\boldsymbol{W}-\boldsymbol{Z}^{(t)}\|_F^2+\frac{\alpha}{L}\|\boldsymbol{W}\|_1\end{aligned} \tag{3.17}$$

在加速的近端梯度算法中，当序列 b_t 满足 $b_t^2 - b_t \leqslant b_{t-1}^2$ 时，令 $\boldsymbol{W}_t$ 是 $\boldsymbol{W}$ 的 t-th 迭代，则有

$$\boldsymbol{W}^{(t)} = \boldsymbol{W}_t + \frac{b_{t-1} - 1}{b_t}(\boldsymbol{W}_t - \boldsymbol{W}_{t-1})$$

能提高算法收敛率到 $O(1/t^2)$[99]。在式(3.17)中，近端梯度联合 $g(\boldsymbol{W})$ 是一个软阈值操作，也就是说，在每次迭代中，$\boldsymbol{W}^*$ 能被获得通过下面的优化问题：

$$\boldsymbol{W}^{(t+1)} = \text{pro}x_\varepsilon[\boldsymbol{Z}^{(t)}] = \underset{\boldsymbol{W}}{\text{argmin}} \frac{1}{2} \| \boldsymbol{W} - \boldsymbol{Z}^{(t)} \|_F^2 + \varepsilon \| \boldsymbol{W} \|_1 \tag{3.18}$$

式中：$\text{pro}x_\varepsilon[\cdot]$ 是一个软阈值操作，可定义为

$$\text{pro}x_\varepsilon[w_{ij}] = \begin{cases} w_{ij} - \varepsilon, & w_{ij} > \varepsilon \\ w_{ij} + \varepsilon, & w_{ij} < \varepsilon \\ 0, & \text{其他} \end{cases} \tag{3.19}$$

根据式(3.17)和式(3.19)，$\boldsymbol{W}$ 能被获得通过下面的软阈值操作：

$$\boldsymbol{W}^{(t+1)} = \text{pro}x_{\frac{\alpha}{L}}[\boldsymbol{Z}^{(t)}] \tag{3.20}$$

根据上述的描述，提出的 MLWSE-L1 可描述为算法 3.1。

算法 3.1　MLWSE-L1

输入：

训练集矩阵 $\boldsymbol{X} \in \mathbb{R}^{n \times d}$；标签矩阵 $\boldsymbol{Y} \in \mathbb{R}^{n \times l}$，基学习器 $\{C_i\}_{i=1}^m$；参数 α、β、η；

输出：

权重矩阵 $\boldsymbol{W}^* \in \mathbb{R}^{ml \times l}$

步骤：

1. 通过基分类器 $\{C_i\}_{i=1}^m$ 生成置信度得分矩阵 $\boldsymbol{S} \in \mathbb{R}^{n \times ml}$
2. 初始化 $b_0, b_1 \leftarrow 1$；$t \leftarrow 1$；$\boldsymbol{W}_0, \boldsymbol{W}_1 \leftarrow (\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S} + \eta \boldsymbol{I})^{-1}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{Y}$
3. 计算矩阵 $\boldsymbol{Y}$ 的拉普拉斯矩阵 $\boldsymbol{H}$
4. 根据式(3.16)计算 L
5. **while** not converged **do**
6. $\quad \boldsymbol{W}^{(t)} \leftarrow \boldsymbol{W}_t + \frac{b_{t-1} - 1}{b_t}(\boldsymbol{W}_t - \boldsymbol{W}_{t-1})$
7. $\quad$ 根据式(3.15)计算 $\nabla f(\boldsymbol{W}^{(t)})$
8. $\quad \boldsymbol{Z}^{(t)} \leftarrow \boldsymbol{W}^{(t)} - \frac{1}{L}\nabla f(\boldsymbol{W}^{(t)})$
9. $\quad \boldsymbol{W}^{(t+1)} \leftarrow \text{pro}x_{\frac{\alpha}{L}}[\boldsymbol{Z}^{(t)}]$
10. $\quad b_{t+1} \leftarrow \frac{1 + \sqrt{4b_t^2 + 1}}{2}$
11. $\quad t \leftarrow t + 1$
12. **return** $\boldsymbol{W}^* \leftarrow \boldsymbol{W}^{(t+1)}$

3.4.2　MLWSE-L21 优化

使用块坐标下降算法优化 MLWSE-L21，块坐标下降可以分为两部分：一是不同特征组之间的外循环；二是每个子块的内循环[103]。在我们的方法，置信度得分矩阵 $\boldsymbol{S}$ 可以分为 m 个组，即 $\boldsymbol{S}^1,\boldsymbol{S}^2,\cdots,\boldsymbol{S}^m$。当 $\boldsymbol{S}$ 是第 k 组，令 $\boldsymbol{S}^{-k}$ 表示余下的组，同理，$\boldsymbol{W}^{-k}$ 是权重 $\boldsymbol{W}$ 余下组的权重。当选择第 k 组时，其他组被固定，则目标函数仅仅最小化 $\boldsymbol{W}^k$，因此在每个块可以最小化如下目标：

$$\frac{1}{2}\|r_{-k}-\boldsymbol{S}^{(k)}\boldsymbol{W}^{(k)}\|_2^2+(1-\alpha)\lambda c_k\|\boldsymbol{W}^{(k)}\|_2+\alpha\lambda\|\boldsymbol{W}^{(k)}\|_1+\frac{\beta}{2}\mathrm{tr}(\boldsymbol{W}^{(k)}\boldsymbol{H}\boldsymbol{W}^{(k)\mathrm{T}}) \tag{3.21}$$

式中：r_{-k} 表示除了组 k 之外，标签 $\boldsymbol{Y}$ 的部分残差，即

$$r_{-k}=\|\boldsymbol{Y}-\sum_{j\neq k}\boldsymbol{S}^{(j)}\boldsymbol{W}^{(j)}\| \tag{3.22}$$

令 $\ell(\boldsymbol{r}_{-k},\boldsymbol{W}^{(k)})=\frac{1}{2}\|r_{-k}-\boldsymbol{S}^{(k)}\boldsymbol{W}^{(k)}\|_2^2$ 表示最小二乘损失函数，其梯度为 $\nabla\ell(r_{-k},\boldsymbol{W}^{(k)})$。我们的目标是最小化式(3.21)获得最优权重 $\boldsymbol{W}_*^{(k)}$，设优化中心点为 $\boldsymbol{W}_0^{(k)}$，t 为优化步，优化目标式(3.21)等价于优化下面的函数：

$$\frac{1}{2t}\|\boldsymbol{W}^{(k)}-(\boldsymbol{W}_0^{(k)}-t\nabla\ell(r_{-k},\boldsymbol{W}_0^{(k)}))\|_2^2+(1-\alpha)\lambda c_k\|\boldsymbol{W}^{(k)}\|_2+\alpha\lambda\|\boldsymbol{W}^{(k)}\|_1+\frac{\beta}{2}\mathrm{tr}(\boldsymbol{W}^{(k)}\boldsymbol{H}\boldsymbol{W}^{(k)\mathrm{T}}) \tag{3.23}$$

当 $\boldsymbol{W}_*^{(k)}=\boldsymbol{0}$ 时，必须满足条件[103]

$$\|\zeta(\boldsymbol{W}_0^{(k)}-t\nabla\ell(r_{-k},\boldsymbol{W}_0^{(k)}),t\alpha\lambda)\|_2\leqslant t(1-\alpha)\lambda c_k \tag{3.24}$$

否则，满足条件

$$\left(1-\frac{t(1-\alpha)\lambda c_k}{\|\zeta(\boldsymbol{W}_0^{(k)}-t\nabla\ell(r_{-k},\boldsymbol{W}_0^{(k)}),t\alpha\lambda)\|_2}\right)_+\zeta(\boldsymbol{W}_0^{(k)}-t\nabla\ell(r_{-k},\boldsymbol{W}_0^{(k)}),t\alpha\lambda) \tag{3.25}$$

式中：$\zeta(\cdot)$表示软阈值操作，即

$$(\zeta(z,t\alpha\lambda))_i=\mathrm{sign}(z_i)(|z_i|-t\alpha\lambda)_+ \tag{3.26}$$

内循环能够使用近端梯度进行加速[104]，因此设置 $t=\frac{1}{L}$，其中 L 是 Lipschitz 常数，可以通过式(3.16)获得。详细的基于块坐标下降优化的 MLWSE-L21 算法

可描述为算法 3.2。

算法 3.2 MLWSE-L21

输入：
训练集矩阵 $\boldsymbol{X}\in\mathbb{R}^{n\times d}$；标签矩阵 $\boldsymbol{Y}\in\mathbb{R}^{n\times l}$，基学习器 $\{C_i\}_{i=1}^{m}$；参数 α、β、λ、η；
输出：
权重矩阵 $\boldsymbol{W}^{*}\in\mathbb{R}^{ml\times l}$
步骤：
1. 通过基分类器 $\{C_i\}_{i=1}^{m}$ 生成置信度得分矩阵 $\boldsymbol{S}\in\mathbb{R}^{n\times ml}$
2. 计算矩阵 $\boldsymbol{Y}$ 的拉普拉斯矩阵 $\boldsymbol{H}$
3. 根据式(3.16)计算 L
4. 根据式(3.22)计算 r_{-k}
5. 循环迭代每个组；为每个组 k，执行步骤 6
6. 初始化 $t\leftarrow 1/L$，$\boldsymbol{W}^{(k)}\leftarrow(\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S}+\eta\boldsymbol{I})^{-1}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{Y}$
7. 根据式(3.24)判断是否 $\boldsymbol{W}^{(k)}=\boldsymbol{0}$，否则执行步骤 8
8. **while** not converged **do**
9. 更新梯度 $\nabla\ell(r_{-k},\boldsymbol{W}^{(k)})$
10. 根据式(3.25)更新权重 $\boldsymbol{W}^{(k+1)}$
11. **return** $\boldsymbol{W}^{*}\leftarrow\boldsymbol{W}^{(t+1)}$

3.5 实验结果与分析

本实验采用 2D 仿真数据集、Benchmark 基准数据集和真实的心脑血管疾病数据集这三种多样化的数据集来评估 MLWSE 与其他同类算法的性能。在实验过程中，本书运用了六种不同的度量标准来全面比较各算法的表现。此外，还深入分析了 MLWSE 算法在鲁棒性、参数敏感性和收敛性等多个关键维度上的性能特点。

3.5.1 2D 仿真实验

基于不同的分布场景设计了 2D 的合成实验，旨在评估算法分类器选择能力。由于多标签分类能被转化为多个二值的分类器问题，这里只考虑单标签的场景。如图 3.3 所示，单变量 X 属于均匀分布 $[-4,4]$ 区间，令 $I(\cdot)$ 表示指示函数，$N(0,1)$ 是标准正态分布，则四种场景生成函数[105]如下：

场景 1：$Y=-2\times I(X<-3)+2.55\times I(X>-2)-2\times I(X>0)+4\times I(X>2)-1\times I(X>3)+N(0,1)$

场景 2：$Y=5+0.4X-0.36X^2+0.005X^3+N(0,1)$

场景 3：$Y=2.85\times\sin(\frac{\pi}{2}\times X)+N(0,1)$

场景 4：$Y=3.85\times\sin(3\pi\times X)\times I(X>0)+N(0,1)$

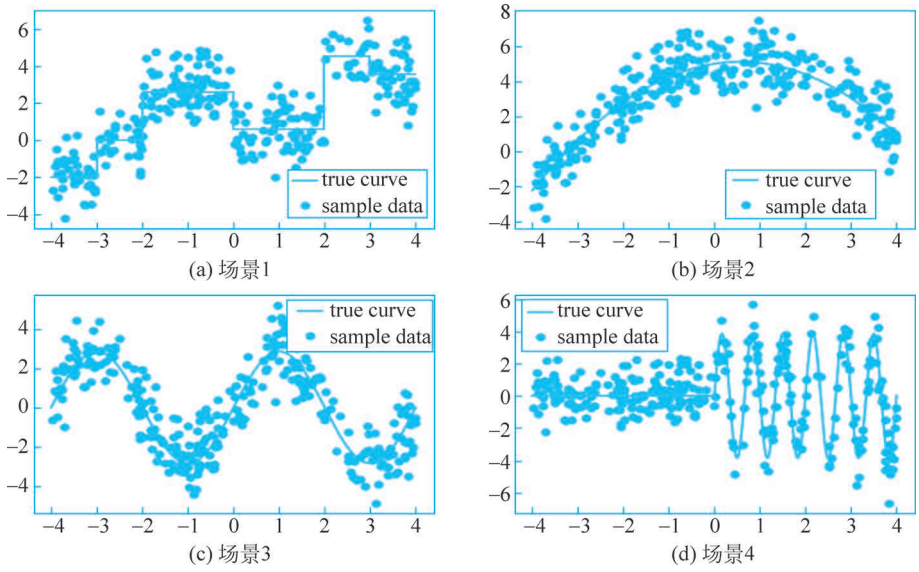


图 3.3　四种场景

注：曲线表示真实的分布，仿真样本数为 300。

使用消融研究评估 MLWSE 算法的分类选择能力，包括基于式(3.3)的 Baseline 选择、式(3.5)的 Lasso 选择、式(3.6)的 Group Sparsity Lasso 选择，随机地划分 35%为训练集、35%为验证集、30%为测试集，随机开展实验 5 次并求平均，4 种场景实验结果如表 3.1 所列。在场景 1 中，三个基分类器都获得了好的结果，但是基于多项式核的 SVM 效果是最好的，Lasso 选择和 Group Lasso 选择分别获得了 0.522 和 0.533 的结果，对应的权重向量为分别为<0.339，0.589，0.052>和<0，0.933，0.045>，能够看出提出的方法能够给好的基分类器分配较高的权重。在场景 2 中，基于多项式核的 SVM 是最优的基分类器，同样地，Lasso 选择和 Group Lasso 选择也是指派较高的权重给最优的分类器，同样的趋势在场景 3 和场景 4 中一样存在。在实际中，各自最优的分类器是不知道的，提出的方法能自适应地给最优的基分类器分类最优权重，并且适应真实场景的改变，实验结果指出了提出方法的加权分类器选择是有效的。

表 3.1　实验结果在 2D 仿真的数据集

Algorithms		场景 1		场景 2		场景 3		场景 4	
		Acc	*W*	Acc	*W*	Acc	*W*	Acc	*W*
Base classifier	SVM (linear kernel)	0.522	—	0.444	—	0.533	—	0.767	—
	SVM (poly kernel)	0.533	—	0.467	—	0.533	—	0.767	—
	Random Forest	0.522	—	0.467	—	0.833	—	0.711	—
Baseline	SVM (linear kernel)		−0.679		−0.927		−59.118		−30.117
	SVM (poly kernel)	0.488	1.620	0.500	2.186	0.800	56.915	0.767	30.061
	Random Forest		0.036		−0.197		0.937		0.164
Lasso selection	SVM (linear kernel)		0.339		−0.852		−0.018		0.072
	SVM (poly kernel)	0.522	0.589	0.500	2.105	0.833	0.105	0.767	0.660
	Random Forest		0.052		−0.197		0.920		0.277
Group sparsity lasso selection	SVM (linear kernel)		0		0		0		0
	SVM (poly kernel)	0.533	0.933	0.5111	1.172	0.833	0.080	0.767	0.732
	Random Forest		0.045		−0.181		0.923		0.274

3.5.2　Benchmark 基准实验

为了验证提出方法的性能，我们在 13 个公开的多标签基准数据集上进行了算法比较，数据集总结在表 3.2。

表 3.2　Benchmark 基准数据集

Datasets	Domain	Instances	Features	Labels	LC
Emotions	Music	593	72	6	1.868
Flags	Image	194	19	7	3.392
Scene	Image	2407	294	6	1.074
Yeast	Biology	2417	103	14	4.237
Birds	Audio	645	260	19	1.014
GpositiveGO	Biology	519	912	4	1.008
CHD-49	Medicine	555	49	6	2.580
Enron	Text	1702	1001	53	3.378
Langlog	Text	1460	1004	75	1.180
Medical	Text	978	1449	45	1.245
VirusGo	Biology	207	749	6	1.217
Water-qy	Chemistry	1060	16	14	5.073
3s-bbc1000	Text	352	1000	6	1.125

注：LC 表示每个样本所属标签的平均数，$\mathrm{LC}=\frac{1}{N}\sum_{i=1}^{n}|Y_i|$。

针对基准数据集，本书比较了 7 种优秀的多标签集成方法，涵盖了传统多标签集成算法的大部分，如 EBR[9]、ECC[9]、EPS[10]、RAkEL[11]、CDE[12]、AdaBoost.MH[15]、MLS[17]等，这些方法都被实现基于 Mulan[106]库和 Meka[107]库。对于 MLWSE，置信度得分矩阵 $\boldsymbol{S}$ 被生成基于基学习器 BR[24]、CC[9]、LP[25]，相关参数使用默认的 scikit-multilearn[108]库设置。对于 MLWSE-L1，参数 α、β、η 分别设置为 10^{-4}、10^{-3}、0.1；对于 MLWSE-L21，参数 α、λ、β、η 分别设置为 0.05、10^{-3}、10^{-2}、0.1，通过使用 5 折交叉验证进行比较实验，在标签数较为大的数据集，有些算法并不能得到结果，被标记为“DNF”，算法中最好的结果已经被加粗，详细的实验结果在表 3.3 和表 3.4 中。

表 3.3　Benchmark 数据集比较结果（Accuracy、Hamming loss 和 Ranking loss）

Datasets	Accuracy ↑								
	EBR	ECC	EPS	RAkEL	CDE	AdaBoost. MH	MLS	MLWSE-L1	MLWSE-L21
Emotions	0.517±0.034	0.532±0.039	0.533±0.021	0.422±0.028	0.524±0.035	0.028±0.016	0.422±0.028	0.806±0.007	**0.807±0.007**
Flags	0.598±0.067	0.630±0.067	0.590±0.063	0.607±0.051	0.609±0.077	0.514±0.064	0.607±0.051	0.727±0.014	**0.743±0.014**
Scene	0.605±0.008	0.659±0.013	0.642±0.007	0.534±0.017	0.538±0.004	0.000±0.000	0.534±0.017	**0.917±0.001**	0.915±0.003
Yeast	0.489±0.014	0.505±0.008	0.491±0.015	0.434±0.012	0.478±0.008	0.335±0.015	0.434±0.012	**0.804±0.002**	0.801±0.002
Birds	0.593±0.021	0.602±0.018	0.589±0.015	0.568±0.036	0.588±0.039	0.456±0.015	0.568±0.036	0.949±0.003	**0.955±0.002**
GpositiveGO	0.933±0.011	0.929±0.016	0.937±0.008	0.930±0.017	0.928±0.018	0.000±0.000	0.930±0.017	**0.971±0.003**	**0.971±0.005**
CHD-49	0.515±0.02	0.533±0.025	0.531±0.022	0.470±0.018	0.490±0.031	0.464±0.008	0.470±0.018	**0.706±0.011**	0.703±0.013
Enron	0.425±0.015	0.467±0.019	0.376±0.020	0.414±0.012	0.411±0.013	0.151±0.009	0.414±0.012	0.953±0.001	**0.954±0.000**
Langlog	0.232±0.027	0.237±0.023	0.231±0.024	0.250±0.026	DNF	0.142±0.022	0.084±0.019	0.820±0.003	**0.830±0.001**
Medical	0.755±0.024	0.767±0.025	0.754±0.024	0.752±0.033	0.718±0.040	0.000±0.000	0.752±0.033	0.986±0.001	**0.987±0.000**
VirusGo	0.861±0.058	0.859±0.056	0.872±0.043	0.861±0.058	0.872±0.058	0.000±0.000	0.861±0.058	**0.956±0.003**	**0.956±0.005**
Water-qy	0.393±0.007	0.414±0.010	0.204±0.019	0.318±0.010	0.402±0.006	0.157±0.03	0.374±0.007	**0.715±0.004**	0.707±0.007
3s-bbc1000	0.044±0.01	0.123±0.027	0.195±0.027	0.144±0.027	0.144±0.019	0.000±0.000	0.144±0.027	0.805±0.006	**0.810±0.005**

续表

Datasets	Hamming loss ↓								
	EBR	ECC	EPS	RAkEL	CDE	AdaBoost. MH	MLS	MLWSE-L1	MLWSE-L21
Emotions	0.197±0.015	0.205±0.016	0.211±0.015	0.264±0.018	0.212±0.019	0.306±0.010	0.264±0.018	0.194±0.007	**0.193±0.007**
Flags	0.249±0.044	**0.243±0.045**	0.258±0.041	0.253±0.036	0.258±0.052	0.278±0.026	0.253±0.036	0.273±0.014	0.257±0.014
Scene	0.093±0.003	0.094±0.004	0.099±0.003	0.135±0.007	0.136±0.003	0.179±0.002	0.135±0.007	**0.083±0.001**	0.085±0.003
Yeast	0.205±0.006	0.210±0.004	0.212±0.007	0.248±0.008	0.228±0.006	0.232±0.007	0.249±0.008	**0.197±0.002**	0.199±0.002
Birds	**0.042±0.003**	0.043±0.004	0.046±0.002	0.051±0.006	0.047±0.006	0.053±0.002	0.051±0.006	0.051±0.003	0.045±0.001
GpositiveGO	**0.027±0.004**	0.030±0.009	0.031±0.005	**0.027±0.006**	0.031±0.009	0.255±0.007	**0.027±0.006**	0.029±0.003	0.029±0.005
CHD-49	0.299±0.013	0.304±0.020	0.307±0.016	0.325±0.013	0.323±0.022	0.307±0.004	0.325±0.013	**0.294±0.011**	0.297±0.013
Enron	0.048±0.001	0.048±0.002	0.052±0.002	0.051±0.001	0.051±0.001	0.062±0.001	0.051±0.001	0.047±0.001	**0.046±0.000**
Langlog	**0.016±0.001**	**0.016±0.001**	**0.016±0.001**	0.020±0.002	DNF	**0.016±0.001**	0.037±0.002	0.180±0.003	0.170±0.001
Medical	**0.010±0.001**	**0.010±0.001**	0.012±0.001	**0.010±0.001**	0.012±0.001	0.028±0.001	**0.010±0.001**	0.014±0.001	0.013±0.000
VirusGo	0.045±0.012	0.045±0.014	0.047±0.019	**0.042±0.017**	**0.042±0.019**	0.203±0.013	**0.042±0.017**	0.044±0.003	0.044±0.005
Water-qy	0.293±0.009	0.295±0.009	0.323±0.002	0.329±0.004	0.303±0.010	0.338±0.008	0.311±0.005	**0.286±0.004**	0.293±0.007
3s-bbc1000	0.209±0.011	0.223±0.012	0.206±0.010	0.251±0.029	0.250±0.013	**0.188±0.008**	0.251±0.029	0.195±0.006	0.190±0.005

续表

Datasets	Ranking loss ↓								
	EBR	ECC	EPS	RAkEL	CDE	AdaBoost. MH	MLS	MLWSE-L1	MLWSE-L21
Emotions	0.171±0.019	0.171±0.013	0.196±0.015	0.316±0.031	0.176±0.019	0.427±0.029	0.326±0.036	0.159±0.013	**0.149±0.011**
Flags	0.201±0.032	0.217±0.041	0.220±0.051	0.318±0.042	0.256±0.060	0.238±0.034	0.272±0.035	0.233±0.021	**0.200±0.011**
Scene	0.079±0.009	0.092±0.009	0.101±0.008	0.195±0.015	0.138±0.010	0.472±0.013	0.227±0.021	**0.068±0.003**	0.069±0.003
Yeast	0.185±0.010	0.191±0.010	0.202±0.008	0.336±0.015	0.219±0.009	0.363±0.029	0.316±0.012	0.171±0.001	**0.168±0.001**
Birds	**0.098±0.012**	0.111±0.013	0.140±0.014	0.199±0.026	0.134±0.015	0.229±0.037	0.168±0.012	0.120±0.008	0.110±0.003
GpositiveGO	0.025±0.005	0.027±0.008	0.031±0.011	0.034±0.012	0.029±0.012	0.301±0.019	0.025±0.006	0.026±0.005	**0.024±0.004**
CHD-49	0.222±0.015	0.230±0.020	0.226±0.021	0.313±0.014	0.255±0.027	0.222±0.011	0.313±0.020	0.215±0.006	**0.210±0.007**
Enron	**0.085±0.008**	0.150±0.014	0.161±0.011	0.302±0.011	0.198±0.001	0.240±0.011	0.175±0.005	0.105±0.003	0.092±0.007
Langlog	**0.121±0.005**	0.273±0.017	0.291±0.013	0.413±0.011	DNF	0.470±0.015	0.166±0.039	0.248±0.005	0.230±0.004
Medical	0.031±0.003	0.042±0.011	0.057±0.011	0.097±0.016	0.074±0.005	0.285±0.010	0.070±0.016	0.033±0.009	**0.025±0.004**
VirusGo	**0.030±0.015**	0.033±0.015	**0.030±0.017**	0.067±0.055	0.043±0.018	0.264±0.045	0.042±0.025	0.031±0.004	0.032±0.005
Water-qy	0.253±0.006	0.256±0.006	0.347±0.007	0.368±0.007	0.275±0.005	0.374±0.011	0.325±0.006	**0.247±0.008**	0.262±0.006
3s-bbc1000	0.404±0.034	0.417±0.031	0.383±0.037	0.497±0.035	0.434±0.003	0.422±0.027	0.497±0.058	**0.381±0.020**	0.389±0.025

表 3.4　Benchmark 数据集比较结果(F1、Macro-F1 和 Micro-F1)

Datasets	F1 ↑								
	EBR	ECC	EPS	RAkEL	CDE	AdaBoost. MH	MLS	MLWSE-L1	MLWSE-L21
Emotions	0.597±0.037	0.612±0.037	0.615±0.018	0.509±0.036	0.608±0.031	0.037±0.02	0.509±0.036	**0.639±0.024**	0.614±0.014
Flags	0.711±0.057	**0.735±0.050**	0.699±0.049	0.721±0.043	0.721±0.065	0.631±0.063	0.721±0.043	0.700±0.020	0.721±0.025
Scene	0.620±0.007	0.675±0.014	0.655±0.006	0.573±0.016	0.573±0.009	0.000±0.000	0.573±0.016	**0.708±0.005**	0.672±0.010
Yeast	0.599±0.014	0.611±0.007	0.599±0.013	0.556±0.012	0.595±0.007	0.456±0.019	0.556±0.012	**0.647±0.006**	0.625±0.004
Birds	0.618±0.022	**0.631±0.016**	0.616±0.019	0.603±0.037	0.621±0.04	0.456±0.015	0.603±0.037	0.152±0.024	0.140±0.009
GpositiveGO	0.938±0.012	0.931±0.017	0.940±0.008	0.934±0.018	0.933±0.018	0.000±0.000	0.934±0.018	**0.945±0.009**	0.941±0.008
CHD-49	0.628±0.022	0.643±0.024	0.643±0.016	0.587±0.016	0.610±0.032	0.580±0.007	0.587±0.016	**0.659±0.008**	0.654±0.016
Enron	0.537±0.015	**0.579±0.017**	0.472±0.020	0.525±0.012	0.523±0.012	0.231±0.013	0.525±0.012	0.578±0.011	0.576±0.006
Langlog	0.239±0.026	0.246±0.020	0.236±0.024	0.267±0.025	DNF	0.142±0.022	0.115±0.026	0.487±0.004	**0.496±0.002**
Medical	0.785±0.025	**0.795±0.026**	0.779±0.024	0.783±0.031	0.751±0.043	0.000±0.000	0.783±0.031	0.773±0.015	0.770±0.011
VirusGo	0.883±0.057	0.879±0.055	0.893±0.037	0.880±0.056	0.893±0.047	0.000±0.000	0.880±0.056	**0.913±0.008**	0.905±0.013
Water-qy	0.532±0.007	0.556±0.011	0.299±0.022	0.452±0.011	0.543±0.006	0.244±0.043	0.513±0.006	0.550±0.009	**0.557±0.011**
3s-bbc1000	0.047±0.012	0.128±0.027	**0.207±0.028**	0.162±0.029	0.159±0.019	0.000±0.000	0.162±0.029	0.051±0.022	0.043±0.021

续表

Datasets	Macro-F1 ↑								
	EBR	ECC	EPS	RAkEL	CDE	AdaBoost. MH	MLS	MLWSE-L1	MLWSE-L21
Emotions	0.639±0.029	**0.641±0.027**	0.631±0.022	0.551±0.039	0.635±0.037	0.038±0.018	0.551±0.039	0.608±0.023	0.584±0.013
Flags	0.657±0.063	0.671±0.086	0.587±0.065	0.658±0.077	0.668±0.077	0.560±0.129	0.658±0.077	0.687±0.024	**0.711±0.025**
Scene	0.709±0.009	**0.728±0.013**	0.707±0.003	0.634±0.015	0.629±0.002	0.000±0.000	0.634±0.015	0.700±0.005	0.665±0.010
Yeast	0.385±0.009	0.398±0.006	0.374±0.005	0.383±0.010	0.405±0.011	0.122±0.003	0.384±0.009	**0.619±0.006**	0.593±0.004
Birds	0.321±0.055	0.291±0.012	0.265±0.052	**0.349±0.048**	0.336±0.057	0.053±0.033	**0.349±0.048**	0.141±0.022	0.133±0.010
GpositiveGO	0.871±0.045	0.854±0.062	0.901±0.047	0.859±0.054	0.845±0.056	0.000±0.000	0.859±0.054	**0.943±0.008**	0.940±0.007
CHD-49	0.498±0.015	0.512±0.026	0.510±0.017	0.470±0.022	0.490±0.030	0.270±0.002	0.470±0.022	**0.629±0.007**	0.624±0.017
Enron	0.219±0.015	0.225±0.016	0.182±0.010	0.214±0.021	0.157±0.000	0.085±0.014	0.214±0.021	**0.548±0.009**	0.547±0.005
Langlog	0.270±0.047	0.273±0.048	0.264±0.043	0.284±0.048	DNF	0.237±0.047	0.051±0.001	0.474±0.006	**0.478±0.003**
Medical	0.653±0.029	0.630±0.031	0.616±0.058	0.669±0.037	0.468±0.002	0.324±0.036	0.669±0.037	**0.758±0.015**	0.755±0.011
VirusGo	0.796±0.078	0.833±0.072	0.844±0.090	0.803±0.069	0.858±0.089	0.067±0.082	0.803±0.069	**0.902±0.009**	0.894±0.011
Water-qy	0.502±0.005	0.523±0.011	0.177±0.019	0.413±0.012	0.503±0.004	0.091±0.020	0.466±0.011	0.518±0.011	**0.526±0.010**
3s-bbc1000	0.062±0.032	0.115±0.027	**0.246±0.028**	0.189±0.051	0.180±0.002	0.000±0.000	0.189±0.051	0.049±0.021	0.036±0.023

续表

Datasets	Micro-F1 ↑								
	EBR	ECC	EPS	RAkEL	CDE	AdaBoost. MH	MLS	MLWSE-L1	MLWSE-L21
Emotions	0.662±0.028	0.663±0.025	0.654±0.023	0.564±0.038	0.654±0.034	0.063±0.032	0.564±0.038	**0.664±0.013**	0.658±0.013
Flags	0.746±0.051	**0.760±0.051**	0.725±0.05	0.745±0.046	0.741±0.063	0.693±0.064	0.745±0.046	0.719±0.017	0.737±0.017
Scene	0.705±0.007	0.718±0.012	0.700±0.006	0.624±0.015	0.617±0.003	0.000±0.000	0.624±0.015	**0.750±0.004**	0.733±0.009
Yeast	0.628±0.011	0.636±0.006	0.625±0.012	0.581±0.012	0.617±0.006	0.480±0.016	0.581±0.011	**0.644±0.006**	0.621±0.004
Birds	0.431±0.054	0.450±0.031	0.402±0.034	0.444±0.048	**0.456±0.055**	0.000±0.000	0.444±0.048	0.365±0.031	0.359±0.027
GpositiveGO	**0.947±0.008**	0.939±0.018	0.939±0.009	0.946±0.013	0.938±0.018	0.000±0.000	0.946±0.013	0.942±0.005	0.942±0.009
CHD-49	0.655±0.017	**0.667±0.025**	0.663±0.018	0.619±0.019	0.638±0.028	0.598±0.004	0.619±0.019	0.658±0.006	0.653±0.017
Enron	0.562±0.004	**0.583±0.013**	0.481±0.016	0.550±0.009	0.544±0.002	0.245±0.014	0.550±0.009	0.565±0.007	0.566±0.004
Langlog	0.159±0.022	0.174±0.012	0.156±0.027	0.191±0.014	DNF	0.000±0.000	0.192±0.029	0.532±0.006	**0.544±0.003**
Medical	0.810±0.016	**0.815±0.024**	0.780±0.028	0.813±0.026	0.781±0.027	0.000±0.000	0.813±0.026	0.754±0.013	0.759±0.007
VirusGo	0.890±0.033	0.890±0.036	0.881±0.047	0.897±0.042	**0.898±0.046**	0.000±0.000	0.897±0.042	0.894±0.008	0.894±0.011
Water-qy	0.563±0.006	**0.585±0.011**	0.304±0.024	0.480±0.010	0.570±0.008	0.259±0.045	0.544±0.008	0.559±0.007	0.557±0.009
3s-bbc1000	0.079±0.023	0.173±0.034	**0.277±0.033**	0.215±0.036	0.208±0.030	0.000±0.000	0.215±0.036	0.086±0.033	0.084±0.042

根据表 3.3 和表 3.4 的实验结果，可以得到以下结论：

（1）与基于 bagging 模式的多标签集成方法比较，如 EBR[9]、ECC[9]、EPS[10]、RAkEL[11]、CDE[12]，在大多数情况下，MLWSE 优于 bagging 的集成模式。原因是 MLWSE 能根据不同的标签，给分类器分配不同的标签权重，并且考虑了标签的成对依赖关系。

（2）与基于 stacking 模式的多标签集成方法比较，如 MLS[17]，在大多数情况下，MLWSE 获得了更好的性能，如 Accuracy 和 F1。原因是不同于 MLS，提出的方法考虑了标签的依赖关系，并且基于不同的标签给不同的基分类器分配了不同的权重。

（3）与基于 boosting 模式的多标签集成方法比较，如 AdaBoost.MH[15]，MLWSE 获得了比 AdaBoost. MH 更好的结果，尽管 AdaBoost. MH 考虑了分类器的权重，但是和 MLWSE 权重设置策略不一样，MLWSE 更多地考虑了多标签存在的客观问题，即标签之间的依赖关系。

3.5.3 Real-world 数据实验

为了分析 MLWSE 在实际多标签场景中的应用，本书应用 MLWSE 到真实的心脑血管疾病数据集。数据集来自××省××人民医院，患有心脑血管疾病的病人，样本总数为 3823 个，有 59 个特征和 9 个标签，9 个标签依次是脑缺血性卒中（CIS）、脑出血（CH）、蛛网膜下腔出血（SAH）、脑静脉血栓形成（CVT）、颅内动脉瘤（IA）、脑血管畸形（CVM）、心脏病（HD）、糖尿病（DM）、高血压（HT），标签样本数及标签频率见表 3.5，实验结果见表 3.6。

表 3.5 Real-world 心脑血管疾病数据集

Label	Instances	Label Frequency
CIS	3380	0.884
CH	140	0.036
SAH	134	0.035
CVT	8	0.002
IA	23	0.006
CVM	20	0.005
HD	1133	0.296
DM	920	0.240
HT	2513	0.657

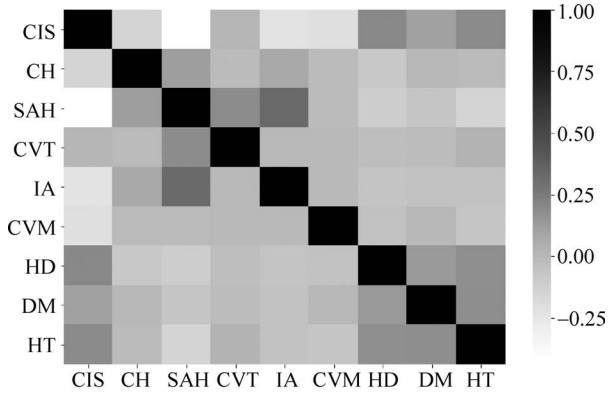
根据表 3.6 的实验结果可知，相比于其他多标签集成方法，在真实的心脑血管疾病数据集，针对不同的评估标准，如 Accuracy、Ranking loss、F1、Macro-F1，提出的方法取得了非常好的实验结果。

进一步，为了验证提出的方法是否考虑了标签的依赖关系，即如果两个标签是

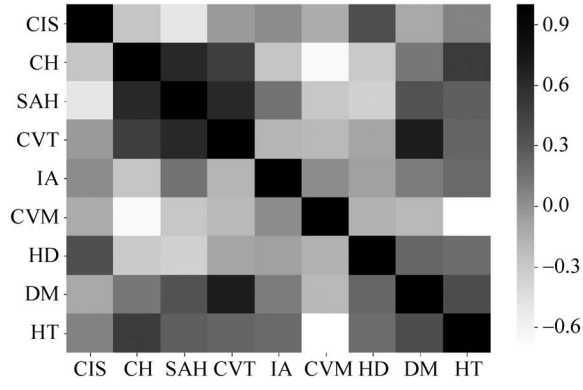
强相关的，那么学习到的权重向量应该也有高的相似，如图 3.4 所示，标签矩阵和权重矩阵存在着很强的灰度表示一致。也就是说，如果标签 y_j 和 y_k 是强的相关，权重向量对（$\boldsymbol{W}_j$，$\boldsymbol{W}_k$）也有高的相似。实验结果表明了提出假设的合理性。

表 3.6　Real-world 心脑血管疾病实验结果

Algorithm	Accuracy ↑	Hamming loss ↓	Ranking loss ↓	F1 ↑	Macro-F1 ↑	Micro-F1 ↑
EBR	0.6923±0.0118	0.0910±0.0050	0.0395±0.0040	0.7694±0.0102	0.4038±0.0464	0.8079±0.0100
ECC	0.7041±0.0082	**0.0896±0.0041**	0.0472±0.0045	0.7800±0.0064	0.4196±0.0495	**0.8156±0.0074**
EPS	0.6904±0.0069	0.0935±0.0034	0.0492±0.0057	0.7673±0.0060	0.4045±0.0508	0.8063±0.0069
RAkEL	0.6797±0.0047	0.0957±0.0028	0.0853±0.0046	0.7597±0.0040	0.3985±0.0477	0.7982±0.0058
CDE	0.6953±0.0060	0.0913±0.0034	0.0579±0.0053	0.7718±0.0049	0.4096±0.0458	0.8094±0.0066
AdaBoost. MH	0.6178±0.0126	0.1201±0.0045	0.0536±0.0044	0.7213±0.0110	0.2146±0.0442	0.7405±0.0094
MLS	0.6797±0.0047	0.0957±0.0028	0.0814±0.0030	0.7597±0.0040	0.3985±0.0477	0.7982±0.0058
MLWSE-L1	**0.9090±0.0015**	0.0910±0.0015	**0.0388±0.0016**	**0.7979±0.0035**	**0.7686±0.0027**	0.8102±0.0038
MLWSE-L21	**0.9101±0.0009**	0.0899±0.0009	**0.0384±0.0078**	**0.7968±0.0035**	**0.7681±0.0033**	0.8116±0.0023



(a) 标签矩阵灰度图



(b) 对应的权重矩阵灰度图

图 3.4　标签矩阵灰度图和对应的权重矩阵灰度图

3.5.4 Friedman 检验分析

本书使用 Friedman 检验[109-110]分析不同算法之间的性能，对每一个算法度量标准，表 3.7 提供了 Friedman 统计 F_F 和 $\alpha=0.05$ 相应的临界值。基于 Friedman 检验分析，当两个算法的性能显著不同时，需要通过“后续检验”来进一步区分各个算法之间的不同，常用的方法有 Nemenyi 后续检验[110]。实验中，通过 Nemenyi 检验计算得到平均序值差别临界值

$$\mathrm{CD}=q_a\sqrt{\frac{k(k+1)}{6N}}$$

当 $\alpha=0.05$ 时，$q_a=3.102$。也就是说，当两个算法的平均序值之差超过临界值 CD 时，则以相应的置信度值认为两个算法的性能有显著的不同。对 Nemenyi 后续检验，使用参数 $k=9$，$N=14$，其中包括 13 个 Benchmark 数据集和 1 个真实的心脑血管疾病数据集，通过计算 CD=3.211。图 3.5 显示了不同评价指标的 CD 图，通过观察每个子图，提出的 MLWSE 算法性能和其他算法有显著的不同。

表 3.7 Friedman 检验 F_F（$k=9$，$N=14$）和对应的不同度量的临界值

Metric	F_F	Critical Value（$\alpha=0.05$）
Accuracy	35.075	3.211
Hamming loss	6.348	
Ranking loss	37.824	
F1	9.261	
Macro-F1	10.243	
Micro-F1	8.312	

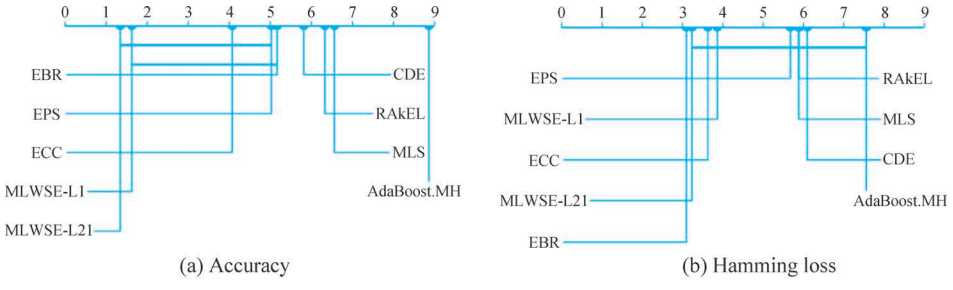


图 3.5 不同评价指标的 CD 图

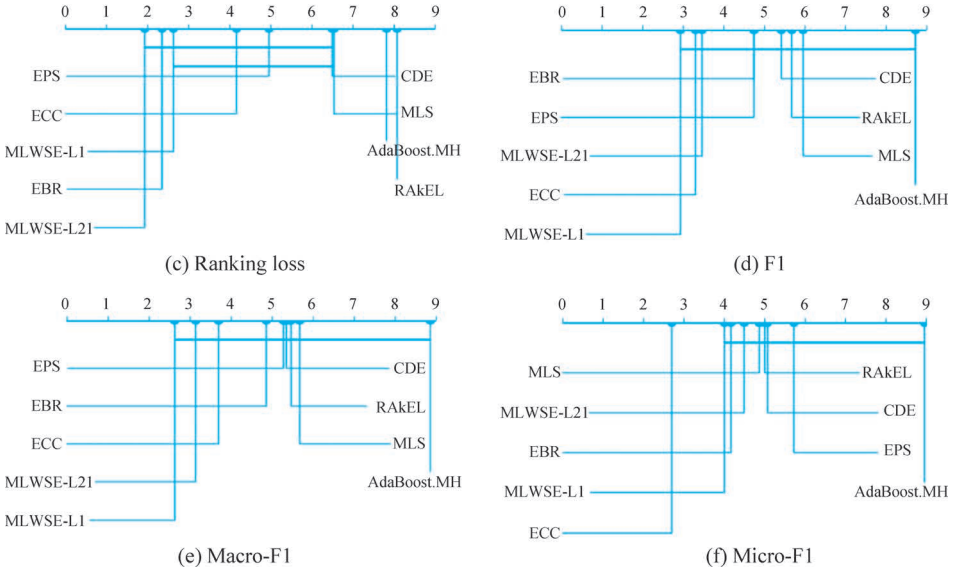


图 3.5　（续）

根据 Friedman 检验结果，可以得出以下结论：

（1）EBR 方法在 Hamming loss 上优于其他方法。因为 EBR 是基于 BR 模型集成的，主要在于优化汉明损失，不考虑标签的相关性。而除 EBR 外，提出的 MLWSE-L21 在 Hamming loss 方面优于其他方法。

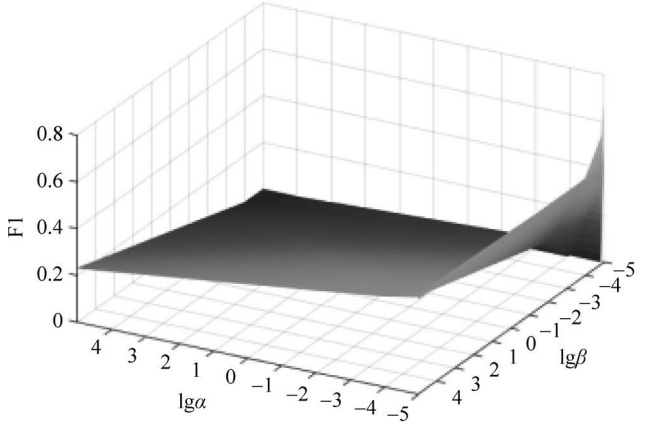
（2）ECC 在 Micro-F1 上优于其他方法。因为 ECC 是一种高阶方法，考虑了标签的全局依赖关系，它试图对全局标签进行建模。而除 ECC 外，提出的 MLWSE-L1 在 Micro-F1 方面优于其他方法。

（3）MLWSE 在其他四个方面都优于相关的多标签集成方法，指出了提出的方法使用局部标签依赖关系和加权分类器选择是有效的。

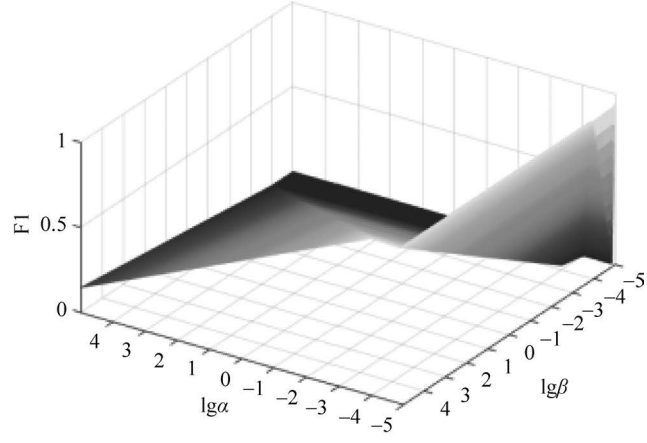
3.5.5　参数敏感性分析

实验在 Emotions 和 GpositiveGO 两个数据集，分析参数的敏感性。对 MLWSE-L1，参数 α、β 取值范围为 $\{10^{-5}, 10^{-4}, \cdots, 10^{3}, 10^{4}\}$，$\eta$ 设置为 0.1；对 MLWSE-L21，参数 α 的取值范围为 $\{0.01, 0.05, 0.1, 0.15, 0.2\}$，$\beta$ 的取值范围为 $\{10^{-4}, 10^{-3}, \cdots, 10^{1}, 10^{2}\}$，$\lambda$ 的取值范围为 $\{10^{-5}, 10^{-4}, \cdots, 10^{1}, 10^{2}\}$，$\eta$ 设置为 0.1。对每个 (α, β) 对，记录 F1 均值，图 3.6 描述了在 Emotions 和 GpositiveGO 两个数据集上 α 和 β 参数的影响。从图 3.6 能够看出：①当 α 取值较大时，MLWSE-L1 的性能较差，尤其是当 $\alpha > 10$ 时，MLWSE-L1 性能很差；②随着 β 值的增加，

MLWSE-L1 性能开始提高，随后下降，因此最终固定参数 α、β 分别在 10^{-4}、10^{-3}。



(a) Emotions数据集



(b) GpositiveGO数据集

图 3.6 MLWSE-L1 参数敏感性分析

在 MLWSE-L21，实验首先在 Emotions 数据集上通过使用 5 折交叉验证选择一组最好的参数，然后保持这个参数不变，改变另外两个进行分析，如图 3.7(a)～(l)所示。通过分析可以看出：

(1) 当固定 α 时，λ 和 β 两个后选集在 $\{10^{-4}, 10^{-3}, 10^{-2}\}$ 能够获得满意的结果；

(2) 当固定 λ 时，取不同的(α, β)值，MLWSE-L21 的性能是稳定的；

(3) 当固定 β 时，α 和 λ 两个后选集在 $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$能够获得满意的结果。

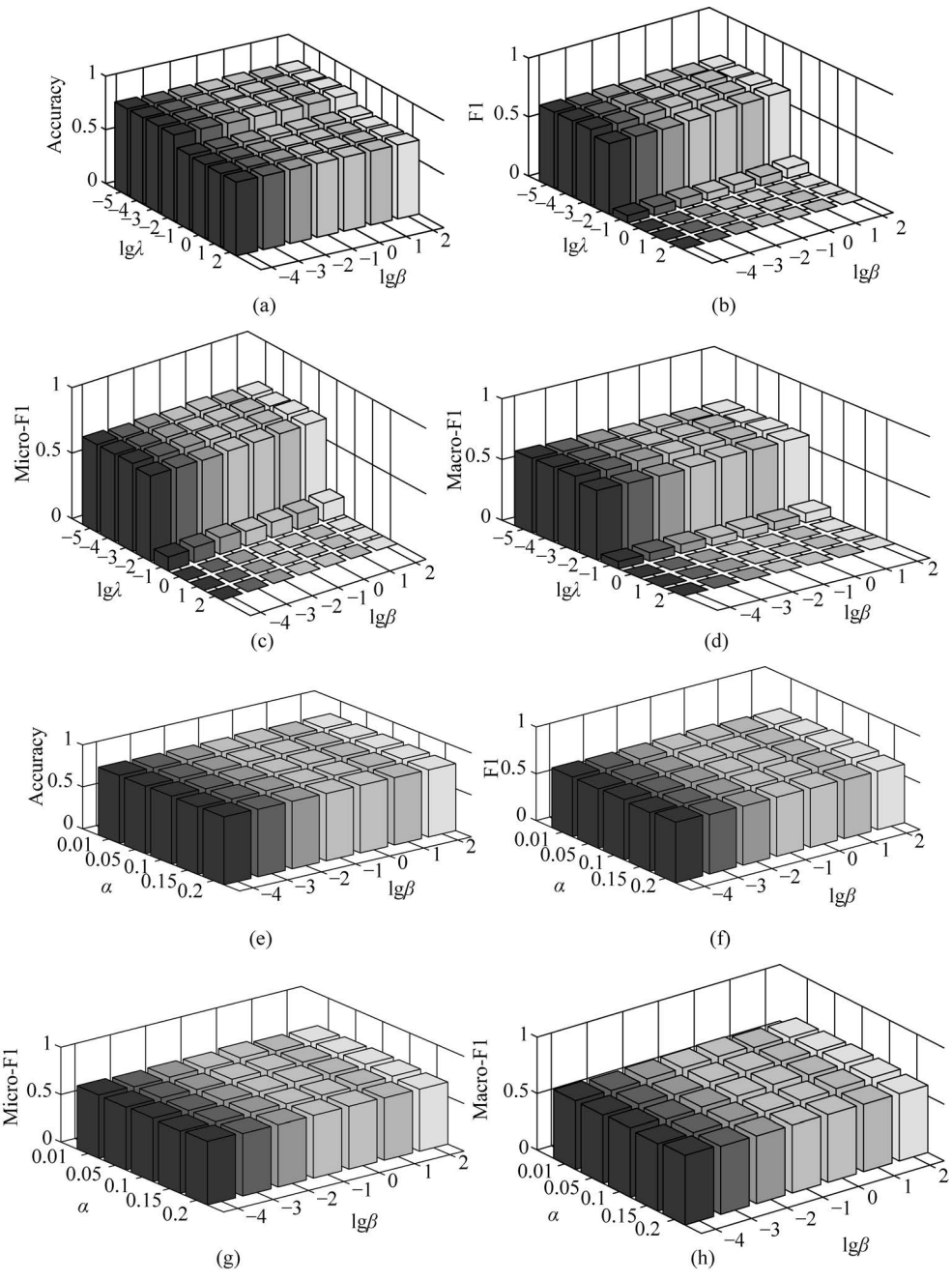


图 3.7　MLWSE-L21 参数敏感性分析

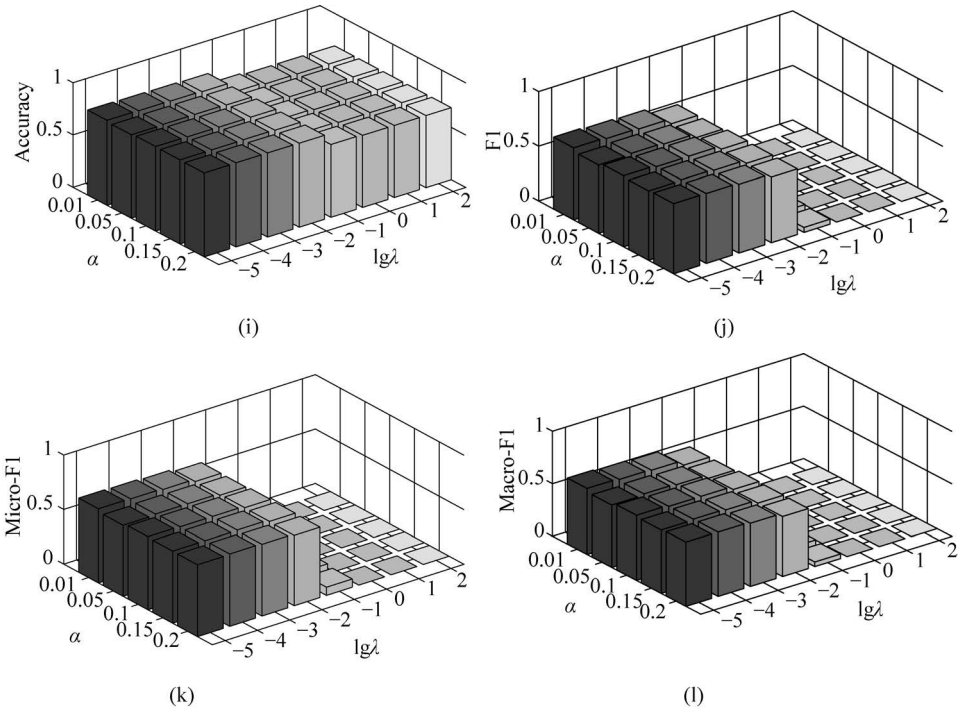


图 3.7 （续）

3.5.6 收敛性分析

分析 MLWSE-L1 和 MLWSE-L21 算法收敛性在 Emotions、Scene、Yeast 和 VirusGo 四个数据集。在 MLWSE 中，使用加速的近端梯度下降和块坐标下降两个算法来优化，加速的近端梯度已经被证明可以收敛到 $O(1/t^2)$[99]，而块坐标下降算法已经被证明可以收敛到 $O((\log t/t)^2)$[101]。图 3.8 显示了随着迭代数的增

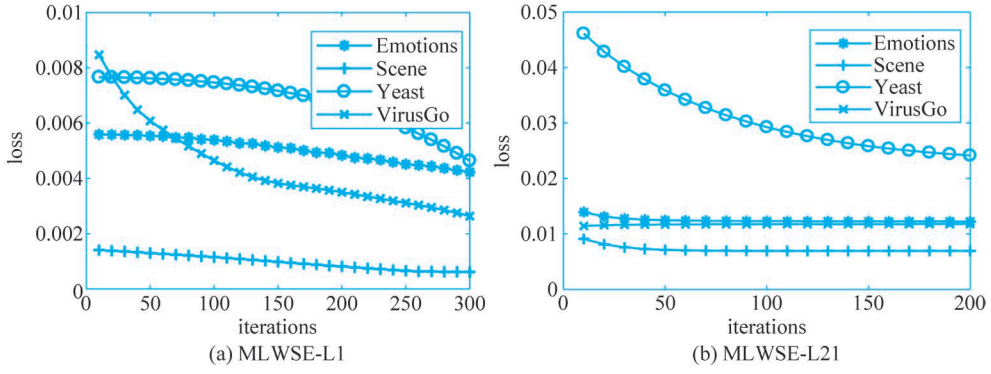


图 3.8 MLWSE 算法收敛性分析

加，MLWSE 损失值的变化情况，图 3.8(a)为 MLWSE-L1，当迭代到 300 时，损失值趋于稳定，当迭代次数到 200 时，损失值降到 0.008，对性能有较小的影响，因此我们实验设置迭代次数为 200；图 3.8(b)为 MLWSE-L21，当外循环迭代次数到 200 时，损失值趋于稳定，而内循环在我们的实验设置为 100，根据实验分析，提出的 MLWSE 有很好的收敛率并且比一些多标签集成方法较快，因为算法使用了加权的分类器选择策略，减少了计算开销。

3.6　本章小结

本章介绍了一种用于多标签分类的加权堆叠选择集成算法 MLWSE，它使用稀疏性进行正则化以促进分类器选择和集成构建，同时利用分类器权重和标签相关性来提高分类性能。另外，提出的 MLWSE 不仅可以作为标签元级特别特征选择方法，而且可以兼容任何现有的多标签分类算法作为其基分类器，最后提出的方法 MLWSE-L1 和 MLWSE-L21 在 13 个多标签基准数据集与真实的心血管和脑血管疾病数据集上进行了综合的实验分析，比较结果证实了算法的优势，且在真实的多标签应用中具有较好的实验结果。