

DOI: 10.11992/tis.201806005

网络出版地址: <http://kns.cnki.net/kcms/detail/23.1538.tp.20190321.0909.002.html>

## 弹性网络核极限学习机的多标记学习算法

王一宾<sup>1,2</sup>, 裴根生<sup>1</sup>, 程玉胜<sup>1,2</sup>

(1. 安庆师范大学 计算机与信息学院, 安徽 安庆 246011; 2. 安徽省高校智能感知与计算重点实验室, 安徽 安庆 246011)

**摘要:** 将正则化极限学习机或者核极限学习机理论应用到多标记分类中, 一定程度上提高了算法的稳定性。但目前这些算法关于损失函数添加的正则项都基于 L2 正则, 导致模型缺乏稀疏性表达。同时, 弹性网络正则化既保证模型鲁棒性且兼具模型稀疏化学习, 但结合弹性网络的极限学习机如何解决多标记问题鲜有研究。基于此, 本文提出一种对核极限学习机添加弹性网络正则化的多标记学习算法。首先, 对多标记数据特征空间使用径向基核函数映射; 随后, 对核极限学习机损失函数施加弹性网络正则项; 最后, 采用坐标下降法迭代求解输出权值以得到最终预测标记。通过对比试验和统计分析表明, 提出的算法具有更好的性能表现。

**关键词:** 多标记学习; 核极限学习机; 正则化; 弹性网络; 径向基函数; 坐标下降法

**中图分类号:** TP391    **文献标志码:** A    **文章编号:** 1673-4785(2019)04-0831-12

中文引用格式: 王一宾, 裴根生, 程玉胜. 弹性网络核极限学习机的多标记学习算法 [J]. 智能系统学报, 2019, 14(4): 831-842.

英文引用格式: WANG Yibin, PEI Gensheng, CHENG Yusheng. Multi-label learning algorithm of an elastic net kernel extreme learning machine[J]. CAAI transactions on intelligent systems, 2019, 14(4): 831-842.

## Multi-label learning algorithm of an elastic net kernel extreme learning machine

WANG Yibin<sup>1,2</sup>, PEI Gensheng<sup>1</sup>, CHENG Yusheng<sup>1,2</sup>

(1. School of Computer and Information, Anqing Normal University, Anqing 246011, China; 2. The University Key Laboratory of Intelligent Perception and Computing of Anhui Province, Anqing 246011, China)

**Abstract:** Regularized extreme learning machine or kernel extreme learning machine theory was applied to multi-label classification, which improves the stability of the algorithm to a certain extent. However, the regularization terms added by these algorithms for loss functions are all based on L2 regularization, which leads to the lack of sparse expression of the model. Simultaneously, elastic net regularization guarantees both model robustness and model sparse learning. Nevertheless, there is insufficient research on how to solve multi-label learning problems by combining elastic net kernel extreme learning machines. Based on this hypothesis, this paper proposes a multi-label learning algorithm that adds elastic network regularization to kernel extreme learning machines. It first uses radial basis function mapping for feature spacing of multi-label; subsequently, it applies the elastic net regularization to the loss function of kernel extreme learning machine. Finally, it uses the coordinate descent method to iteratively solve the output weights to get the final prediction labels. Through comparative experiments and statistical analyses, the proposed method demonstrates better performance.

**Keywords:** multi-label learning; kernel extreme learning machine; regularization; elastic net; radial basis function; coordinate descent

收稿日期: 2018-06-02. 网络出版日期: 2019-03-22.

基金项目: 安徽省高校重点科研项目 (KJ2017A352); 安徽省高校重点实验室基金项目 (ACAIM160102).

通信作者: 程玉胜. E-mail: [chengyusheng@163.com](mailto:chengyusheng@163.com).

近年来, 随着人工智能的迅速发展, 标记学习成为其重点的研究领域之一。其中单标记学习将每个示例由一个特征向量和一个标记来描述; 而

多标记学习<sup>[1]</sup> 则将一个示例同时分配给多个标记,即每个对象由一个特征向量和一个二元标记向量来表示。多标记学习的这种示例表达方式更加契合现实世界对象存在的多义性,因此多标记学习成为模式识别与标记学习的重点研究课题之一,并已成功应用于文本分类<sup>[2-3]</sup>、图像识别<sup>[4]</sup>、生物学习<sup>[5]</sup> 和情感分析<sup>[6]</sup> 等领域。

目前,在多标记学习问题中,诸多学者已研究并提出多种多标记学习算法,而这些方法大致可以分为2类,即问题转换法和算法适应法。其中问题转换法是将多标记学习任务转换为一个或者多个相应单标记学习任务,然后再通过传统单标记学习方法进行处理,典型算法包括BR<sup>[4]</sup>、LP<sup>[7]</sup>、PPT<sup>[8]</sup> 和RAKEL<sup>[9]</sup> 等。而算法适应法通过扩展特定单标记学习算法,修改其约束条件从而可以直接处理多标记学习任务,例如ML-KNN<sup>[10]</sup>、ML-NB<sup>[11]</sup>、Rank-SVM<sup>[12]</sup> 和ML-RBF<sup>[13]</sup> 等。而这些适应型算法就是将最近邻( $k$ -nearest neighbors, KNN)、朴素贝叶斯(naive bayes, NB)、支持向量机(support vector machine, SVM)和径向基函数(radial basis function, RBF)神经网络等算法适应于多标记数据。这些改造的算法在多标记学习中取得了不错的效果。但其中BR、LP、ML-KNN、MLNB和Rank-SVM等算法因本身特点所限,导致其时间消耗较大。

为了解决分类算法时间消耗大的问题,近年间,部分学者提出了多种基于极限学习机(extreme learning machine, ELM)的多标记学习算法。ELM是由Huang等<sup>[14]</sup> 提出的是一种单隐藏层前馈神经网络(single-hidden layer feedforward neural networks, SLFNs)算法,该算法具有模型设计简单、运行速度快和泛化性能高等特点,在多标记学习中具有良好的性能表现。为提高ELM分类模型的稳定性及鲁棒性,邓万宇等<sup>[15]</sup> 提出正则极限学习机算法(regularized extreme learning machine, RELM),对损失函数施加L2惩罚以避免分类模型出现过拟合现象。随后,Miche等<sup>[16]</sup> 提出TROP-ELM(tikhonov-regularized optimally pruned extreme learning machine, TROP-ELM)算法,将L1和L2惩罚级联使用,对隐藏层神经元施加L1惩罚,对回归权重施加L2惩罚,以达到删减神经元个数和稳定数值的作用。但这些算法都需要随机初始化权值和偏置,使得算法对于随机值敏感,导致算法稳定性不高。为处理这一问题,Huang等<sup>[17]</sup> 提出使用核函数映射特征空间以代替传统隐藏层随机特征映射函数,使得该算法

可以直接处理回归问题、单标记和多标记分类。基于ELM的多标记分类算法,ER等<sup>[18]</sup> 和Sun等<sup>[19]</sup> 利用ELM提出一种高速多标记分类器模型,将ELM适应于多标记数据集,分类效果较为理想。Zhang等<sup>[20]</sup> 提出了一种多层ELM-RBF算法,改变传统ELM算法的单隐藏层策略,使用多隐藏层来实现多标记分类,在分类精度上也取得了不错的效果。Luo等<sup>[21]</sup> 首次采用核ELM来处理多标记问题,以保证分类算法的稳定性。对于多标记学习与正则化理论结合部分,Han等<sup>[22]</sup> 提出将多标记学习作为弹性网络惩罚的最小二乘优化问题,并不使用L1惩罚进行稀疏表示。本文创作的思想来源于此,且已有研究表明在多标记数据集中特征之间存在着相关性和冗余性,此时将RELM原L2正则项用弹性网络正则代替,既保证模型稳定性也可对模型进行稀疏性表示。

结合上述ELM算法和正则化理论,本文首次将弹性网络正则结合核极限学习机(kernel extreme learning machine, KELM)应用到多标记分类中,使用弹性网络正则约束核KELM,提出基于弹性网络极限学习机的多标记学习算法(multi-label learning algorithm of elastic net kernel extreme learning machine, ML-EKELM)。该算法通过KELM映射特征空间,然后对损失函数添加弹性网络<sup>[23]</sup> 正则项,最后采用坐标下降法<sup>[24]</sup> 迭代求解多标记目标优化问题。KELM与弹性网络的结合提高了算法鲁棒性,保证了模型稀疏性,提供了一种基于ELM解决多标记问题的新途径。通过对比现有基于ELM的先进多标记算法和经典多标记算法,验证了本文算法的有效性和可靠性。

## 1 基本理论研究

### 1.1 极限学习机理论

传统神经网络算法需要较多的网络参数设置,在求解最优解时很有可能出现局部最优解,而无法得到全局最优解。而极限学习机是一种高效且具有优化学习算法的单隐层前馈神经网络,求解时只需设置隐藏层节点数,并随机初始化权值和偏置就可求解出全局最优解。ELM求解单隐层前馈神经网络,可分为2个阶段:随机特征映射和线性参数求解。

在对ELM两个阶段进行分析之前,需要做出以下形式化定义:设有 $N$ 个随机样本 $\{(X_i, Y_i) | i = 1, 2, \dots, N\}$ ,其中特征空间与标记空间可分别表示为 $X_i = [x_{i1} x_{i2} \dots x_{im}]^T$ ,  $Y_i = [y_{i1} y_{i2} \dots y_{im}]^T$ ,则对于具有 $L$ 个隐藏节点的单隐藏层神经网络形式化定

义为:

$$f_L(X_j) = \sum_{i=1}^L \beta_i g_i(X_j) \quad (1)$$

式中:  $\beta_i = [\beta_{i1} \beta_{i2} \cdots \beta_{im}]^T$  表示输出权值;  $g_i$  表示第  $i$  个隐藏节点的输出, 实质为激活函数, 并可表示为:

$$g_i(X_j) = g(\mathbf{w}_i \cdot \mathbf{X}_j + b_i) \quad (2)$$

式中:  $\mathbf{w}_i = [w_{i1} w_{i2} \cdots w_{im}]^T$  为输入权值;  $b_i$  表示第  $i$  个隐藏神经元的偏置;  $\cdot$  表示为点积。通常式 (1) 用来建模回归, 对于分类问题可使用 sigmoid 函数来限制输出值的范围, 从而达到分类效果。

以上为 ELM 的第 1 阶段即随机特征映射, 对于第 2 阶段的线性参数求解, 通过最小化平方误差的近似误差来求解连接隐藏层和输出层的权值  $\beta$ 。可表示为:

$$\min_{\beta} \|\mathbf{H}\beta - \mathbf{Y}\|^2 \quad (3)$$

式中  $\mathbf{H}$  为隐藏层输出矩阵, 即

$$\mathbf{H} = \begin{bmatrix} h(\mathbf{x}_1) \\ h(\mathbf{x}_2) \\ \vdots \\ h(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & h_2(\mathbf{x}_1) & \cdots & h_L(\mathbf{x}_1) \\ h_1(\mathbf{x}_2) & h_2(\mathbf{x}_2) & \cdots & h_L(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_N) & h_2(\mathbf{x}_N) & \cdots & h_L(\mathbf{x}_N) \end{bmatrix} \quad (4)$$

$\mathbf{Y}$  为训练标记矩阵:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \mathbf{y}_2^T \\ \vdots \\ \mathbf{y}_N^T \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{Nm} \end{bmatrix} \quad (5)$$

通过式 (1)、式 (3), 最小二乘解为

$$\hat{\beta} = \mathbf{H}^+ \mathbf{Y} \quad (6)$$

式中  $\mathbf{H}^+$  表示  $\mathbf{H}$  的 Moore-Penrose 广义逆矩阵, 表示为

$$\text{s.t. } \mathbf{H}^+ = \begin{cases} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T, & \mathbf{H}^T \mathbf{H} \text{ 非奇异} \\ \mathbf{H}^T (\mathbf{H} \mathbf{H}^T)^{-1}, & \mathbf{H} \mathbf{H}^T \text{ 非奇异} \end{cases} \quad (7)$$

最终求出的  $\hat{\beta}$  即可以预测未知标记, 表示为

$$\hat{\mathbf{Y}} = \mathbf{H} \hat{\beta} \quad (8)$$

## 1.2 正则化理论

在机器学习中, 偏差 (bias) 与方差 (variance) 共同影响模型的准确率。高偏差容易导致模型欠拟合 (unfitting), 高方差则会导致模型过拟合 (overfitting)。通常, 解决高偏差可选择使用更为复杂的模型或增加模型参数来降低偏差值, 但这就会导致模型过拟合情况发生。而正则化理论是解决高方差或避免训练模型过拟合的有效方法之一, 在机器学习领域被广泛使用。

通过正则化方式, 可以降低模型的复杂度, 避免可能的过度拟合。近年来, 研究者提出了多种适合机器学习的正则化方法, 其中 L2 正则化、

L1 正则化和弹性网络正则化等使用较为普遍。这些正则化方法详细描述如下:

### 1) L2 正则 (Ridge Regression)

L2 正则化模型 (也称为岭回归) 是在最小化损失函数后添加正则项  $\lambda \|\beta\|_2^2$ , 其中参数  $\lambda \in [0, +\infty]$ , 起到降低权重的作用, 最终得到目标函数:

$$L_2 = \min_{\beta} \|\mathbf{H}\beta - \mathbf{Y}\|_2^2 + \lambda \|\beta\|_2^2 \quad (9)$$

### 2) L1 正则 (Lasso)

L1 正则化模型 (也称为 Lasso 模型) 则是在最小化损失函数添加正则项  $\lambda \|\beta\|_1$ , 其中  $\lambda \in [0, +\infty]$ , 而 Lasso 最大的特点在于产生稀疏权重矩阵, 构造出稀疏模型已达到特征选择的作用, 最终目标函数为

$$L_1 = \min_{\beta} \|\mathbf{H}\beta - \mathbf{Y}\|_2^2 + \lambda \|\beta\|_1 \quad (10)$$

### 3) L2&L1 正则 (Elastic Net)

弹性网络正则化是一种结合 L1 正则与 L2 正则的各自优点的新型正则化方法, 即在最小化损失函数添加正则项  $\lambda(\alpha \|\beta\|_1 + (1-\alpha) \|\beta\|_2^2)$ , 其中  $\lambda \in [0, +\infty]$ ,  $\alpha \in [0, 1]$ , 目标函数定义为

$$L_{\text{ElasticNet}} = \min_{\beta} \|\mathbf{H}\beta - \mathbf{Y}\|_2^2 + \lambda(\alpha \|\beta\|_1 + (1-\alpha) \|\beta\|_2^2) \quad (11)$$

通过式 (11) 可知: 当  $\alpha=0$  时, Elastic Net 即为 L2 正则; 当  $\alpha=1$  时, Elastic Net 即为 L1 正则; 当  $\alpha \in (0, 1)$  时, Elastic Net 将保留 L2 和 L1 正则各自特点, 达到弹性 2 种正则的作用。根据这一特点给出 3 种正则化的二维图像描述, 如图 1 所示。

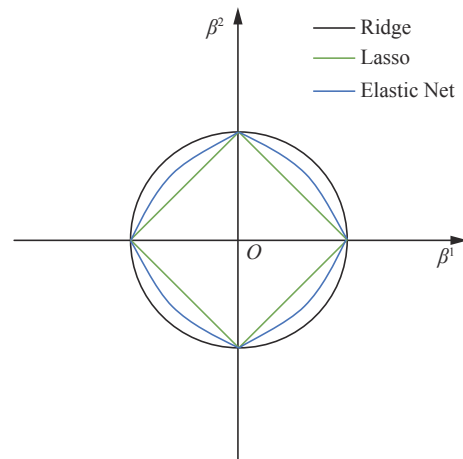


图 1 3 种正则化比较

Fig. 1 The comparison of three regularizations

## 2 基于弹性网络核极限学习机算法

### 2.1 基于 ELM 的多标记学习

传统单标记学习无法对于真实世界对象多语



义性、概念复杂性进行有效处理,且无法满足目前机器学习的高要求,由此建立了多标记学习框架可以解决这一问题。该框架通过对任意一个对象,进行一个特征向量的描述,根据特征向量尽可能将对象进行合适的类别标记和精准分类<sup>[25]</sup>。假定含有  $N$  个样本的多标记数据集,  $\mathbf{X}$  为  $n$  维的示例空间  $\mathbf{R}^n$ ,  $\mathbf{Y}$  为  $m$  类标记空间,则在多标记学习中,给定数据集  $\mathbf{D} = \{(\mathbf{x}_1, \mathbf{Y}_1), (\mathbf{x}_2, \mathbf{Y}_2), \dots, (\mathbf{x}_n, \mathbf{Y}_n)\}$ , 其中  $\mathbf{x}_i \in \mathbf{X}$  是一个示例,  $\mathbf{Y}_i \in \mathbf{Y}$  是一组标记集合  $\{\mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}, \dots, \mathbf{y}_m^{(i)}\}$ , 且  $\mathbf{y}_m^{(i)} \in \mathbf{Y}$ , 可得到映射关系  $f: \mathbf{X} \rightarrow 2^{\mathbf{Y}}$ 。

根据多标记学习的目标,同时结合 ELM 学习模型,ELM 的随机映射函数  $h(\mathbf{x}_i)$  将  $\mathbf{x}_i$  从输入空间映射到  $L$  维的特征空间,  $\mathbf{Y}_i \in \mathbf{R}^m$  为输出标记集合。根据式 (4)、(5) 和 (8),可得多标记 ELM 的输出函数  $f_i(\mathbf{x})$  为:

$$f_i(\mathbf{x}) = \mathbf{H}\boldsymbol{\beta} = \begin{bmatrix} h(\mathbf{x}_1) \\ h(\mathbf{x}_2) \\ \vdots \\ h(\mathbf{x}_N) \end{bmatrix}_{N \times L} \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \boldsymbol{\beta}_2^T \\ \vdots \\ \boldsymbol{\beta}_L^T \end{bmatrix} \quad (12)$$

将式 (9) 和 (12) 结合即为正则 ELM, 即 RELM 算法,该算法添加 L2 正则来提高原始 ELM 算法的稳定性和泛化性能,同时有效避免过拟合,目标函数表示为:

$$\min_{\boldsymbol{\beta}} L_{\text{RELM}} = \frac{1}{2} \|\boldsymbol{\beta}\|^2 + \frac{C}{2} \sum_{i=1}^N \|\boldsymbol{\xi}_i\|^2 \quad (13)$$

式中  $C$  为正则化系数;由式 (6)、(7) 和 (13) 求解目标函数,可得输出权值  $\boldsymbol{\beta}$ :

$$\boldsymbol{\beta} = \mathbf{H}^T \left( \frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{Y} \quad (14)$$

式中  $\mathbf{I}$  为  $L$  维单位矩阵。这样最终多标记学习目标函数表示为:

$$f_i(\mathbf{x}) = \mathbf{H}\boldsymbol{\beta} = \mathbf{H}\mathbf{H}^T \left( \frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{Y} \quad (15)$$

通过式 (14) 得到训练集的输出权值,再通过随机映射函数  $h(\mathbf{x})$  将测试集的特征向量映射,最终多标记预测结果可由式 (15) 得出。这种基于正则化 ELM 的多标记学习算法,不仅在预测精度上取得了不错的效果,并且求解速度也具有较大优势。

## 2.2 弹性核极限学习机的多标记学习算法

基于 L2 正则能够有效控制训练模型过拟合到某个特征上,即假设参数分布服从高斯分布以达到稳定模型的作用;而基于 L1 正则能够约束目标方程稀疏性进而实现特征选择,即假设参数分布服从拉普拉斯分布以保证稀疏化模型。简单来说,即 L2 正则只能让某些参数逼近于 0,而 L1 正

则可以使得某些参数等于 0。基于以上正则化特点分析,结合这 2 种正则的弹性网络可以实现正则和稀疏双重作用<sup>[23]</sup>。本节将使用 Elastic Net 替换原有 RELM 中的 L2 正则,首次运用到多标记学习中。

设多标记数据集  $\mathbf{D} = \{\mathbf{x}_i, \mathbf{Y}_i\}_{i=1}^N$ , 其中  $\mathbf{x}_i \in \mathbf{R}^n$  是  $n$  维特征向量,  $\mathbf{Y}_i \in \mathbf{R}^m$  为输出标记集合。则基于 L2 正则 RELM 替换为 Elastic Net,通过式 (11) 将式 (13) 改写为:

$$\begin{aligned} \min_{\boldsymbol{\beta}} L_E &= \frac{C}{2} \sum_{i=1}^N \|\boldsymbol{\xi}_i\|^2 + \lambda (R_{\alpha}(\boldsymbol{\beta})) \\ R_{\alpha}(\boldsymbol{\beta}) &= \alpha \|\boldsymbol{\beta}\|_1 + (1-\alpha) \|\boldsymbol{\beta}\|_2^2 \\ \text{s.t. } \boldsymbol{\xi}_i &= \mathbf{Y}_i - f_i(\mathbf{x}_i), i = 1, 2, \dots, N \end{aligned} \quad (16)$$

由于传统 ELM 算法需设置隐藏层节点数,并且需初始随机权值和偏置,易受随机值的影响导致计算结果不稳定,采用核 ELM 则可以解决这个问题。根据式 (4) 和 (12),当映射函数  $h(\mathbf{x})$  未知时,即引入核矩阵(本文采用 RBF 核):

$$\begin{aligned} \mathbf{H} &= \boldsymbol{\Omega}_{\text{ELM}} : \boldsymbol{\Omega}_{\text{ELM}(i,j)} = K(\mathbf{x}_i, \mathbf{x}_j) \\ K(\mathbf{x}_i, \mathbf{x}_j) &= \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2) \end{aligned} \quad (17)$$

式中  $\gamma$  一般取值为 1。结合式 (3)、(12) 和 (17),式 (16) 可进一步改写为带有 RBF 核映射的目标函数:

$$\min_{\boldsymbol{\beta}} L_E = \frac{C}{2} \|\mathbf{Y} - \boldsymbol{\Omega}_{\text{ELM}} \boldsymbol{\beta}\|^2 + \lambda R_{\alpha}(\boldsymbol{\beta}) \quad (18)$$

由于 Elastic Net 本身结构特点,具有 L1 正则导致存在不可导点,无法使用类似于 BP 神经网络(back propagation)的梯度下降法(gradient descent)或传统 ELM 将神经网络转化为最小二乘法。坐标下降法<sup>[26]</sup>则可以解决这一问题,因其是一种非梯度优化算法,无需求导目标函数,只需通过坐标方向搜寻最小值,符合弹性网络求解的要求,因此本文采用坐标下降法对于弹性网络核极限学习机进行目标求解。根据式 (18) 求解最小化目标,即

$$\min_{\boldsymbol{\beta}} \frac{C}{2} \sum_{i=1}^N \left( \mathbf{Y}_i - \sum_{j=1}^N \boldsymbol{\Omega}_{\text{ELM}(i,j)} \boldsymbol{\beta}_j \right)^2 + \lambda R_{\alpha}(\boldsymbol{\beta}) \quad (19)$$

通过坐标下降法,式 (19) 的更新公式可表示为:

$$\boldsymbol{\beta}_j \leftarrow \frac{S \left( \sum_{i=1}^N \boldsymbol{\Omega}_{\text{ELM}(i,j)} (\mathbf{Y}_i - \tilde{\mathbf{Y}}_i^{(j)}), \lambda \alpha \right)}{1 + \lambda (1 - \alpha)} \quad (20)$$

式中:  $\tilde{\mathbf{Y}}_i^{(j)}$  为拟合值;  $\mathbf{Y}_i - \tilde{\mathbf{Y}}_i^{(j)}$  为  $\boldsymbol{\beta}_j$  的部分残差;  $S$  是软阈值算子(soft-thresholding)<sup>[27]</sup>,用于处理 L1 惩罚。文献 [28] 给出详细求解算法,最后对于 L2 惩罚进行比例收缩。其中  $S$  详细定义为:

$$S(\delta, \gamma) \equiv \text{sign}(\delta)(|\delta| - \gamma)_+ = \begin{cases} \delta - \gamma, & \delta > 0 \text{ 且 } \gamma < |\delta| \\ \delta + \gamma, & \delta < 0 \text{ 且 } \gamma < |\delta| \\ 0, & \gamma \geq |\delta| \end{cases} \quad (21)$$

在训练集中通过坐标下降法求出输出权值矩阵  $\beta$ , 设  $x_j^*$  为测试数据第  $j$  个示例的特征向量, 则多标记预测结果可以表示为

$$f_i(x_j^*) = \Omega_{\text{ELM}(i,j)} \beta \quad (22)$$

式中:  $\Omega_{\text{ELM}(i,j)}$  表示将训练集全部特征向量与测试集特征向量共同使用式 (17) 的 RBF 核函数映射, 最后提出算法具体步骤如算法 1 所示。

**算法 1** 基于弹性网络核极限学习机的多标记学习算法 (ML-EKELM)。

**输入** 训练数据集  $D = \{x_i, Y_i\}_{i=1}^N$ , 测试数据集  $D^* = \{x_j^*\}_{j=1}^M$ , RBF 核参数  $\gamma$ , 正则化参数  $\lambda$ 、 $\alpha$ , 最大迭代次数  $Q$ ;

**输出** 测试数据集预测标记  $Y^*$ 。

训练 training set  $D$

for training set  $D$

step 1: compute training data kernel matrix  $\Omega_{\text{train}}$  according to Eq. (17);

step 2: calculate the output weight  $\beta$ ;

for  $i=1$  to  $Q$

update output weight  $\beta$  according to Eq. (20) and

Eq. (21)

end for

end for

测试 testing set  $D^*$

for testing set  $D^*$

step 1: compute training and testing data kernel matrix  $\Omega_{\text{test}}$  according to Eq. (17);

step 2: get predicted result  $f_i(x^*)$  according to Eq. (22)

end for

return predicted testing data label set  $Y^* = \{y_j^* | f_i(x_j^*) > 0, j = 1, 2, \dots, M\}$ 。

### 3 实验方案及结果分析

#### 3.1 实验数据描述

为验证本文算法的有效性, 特选取了 Yeast Gene<sup>[12]</sup>、Scene<sup>[4]</sup>、Yahoo Web Pages<sup>[10]</sup> (包含 11 个子数据集) 等 13 个数据集。其中 Yeast Gene 包含 2 417 个样本, 训练数据集有 1 500 个样本, 测试数据集有 917 个样本, 每个样本包含 103 个属性值, 所有的样本大致有 14 种类别属性, 每个样本

对应的平均标记数为 4.24。Scene 数据集由 2 407 张图片组成, 人工手动标记图片 6 类标记, 平均每张图片有  $1.24 \pm 0.44$  个类标记, 特征向量维度为 294 维, 其中 1 211 个训练集和 1 196 个测试集。雅虎网页数据集是从雅虎网站收集, 包括 11 个版块 (“Arts”、“Business”、“Computers”等), 各数据子集特征数在 400~1 100 之间, 各数据子集中包含 2 000 个训练集以及 3 000 个测试集, 详细信息如表 1 所示。

表 1 雅虎网页数据集  
Table 1 Yahoo web pages data set

数据集	特征数	训练平均标记	测试平均标记
Arts	462	1.627±0.882 8	1.642±0.958 9
Business	438	1.590±0.839 8	1.586±0.849 4
Computers	681	1.487±1.086 9	1.522±1.046 8
Education	550	1.465±0.764 2	1.458±0.752 2
Entertainment	640	1.426±0.930 6	1.417±0.991 3
Health	612	1.667±0.851 7	1.659±0.901 3
Recreation	606	1.414±0.813 6	1.429±0.867 9
Reference	793	1.159±0.429 5	1.177±0.536 9
Science	743	1.489±0.787 5	1.425±0.758 0
Social	1047	1.274±0.638 8	1.290±0.617 4
Society	636	1.705±1.206 2	1.684±1.246 4

#### 3.2 多标记评价指标

对于多标记学习, 传统单标记评价指标例如 Accuracy、Precision 和 Recall 都无法直接对多标记学习算法进行指标评价。为有效验证算法综合性能, 本文将使用 5 种多标记通用评价指标进行算法评价, 评价指标包括: Hamming Loss、One-Error、Coverage、Ranking Loss 和 Average Precision<sup>[1]</sup>。

设多标记分类器为  $h(\cdot)$ , 预测函数  $f(\cdot, \cdot)$ , 排序函数  $\text{rank}_f$ 。多标记数据集  $D = \{(x_i, Y_i) | 1 \leq i \leq n\}$ 。上述 5 种评价指标 HL、OE、CV、RL 和 AP 形式化定义如下:

$$\text{HL}_D(h) = \frac{1}{n} \sum_{i=1}^n \frac{1}{|Y_i|} |h(x_i) \Delta Y_i| \quad (23)$$

式中:  $\Delta$  表示两个集合之间的对称差。海明损失是评估对象标记被错误分类标记的次数情况, 正确的标记被错误预测情况。当  $\text{HL}_D(h)=0$  时为最好的情况, 即  $\text{HL}_D(h)$  越小,  $h(\cdot)$  的性能越高。

$$\text{OE}_D(f) = \frac{1}{n} \sum_{i=1}^n [ [\arg \max_{y \in Y} f(x_i, y)] \notin Y_i ] \quad (24)$$

1-错误率是评估对象最高排位标记并未正确

标记的次数情况。当  $OE_D(f)=0$  时为最好的情况, 即  $OE_D(f)$  越小,  $f$  的性能越高。

$$CV_D(f) = \frac{1}{n} \sum_{i=1}^n \max_{y \in Y_i} \text{rank}_f(x_i, y) - 1 \quad (25)$$

覆盖率是评估对象标记序列中所需标记数达到覆盖全部标记, 即  $CV_D(f)$  越小,  $f$  的性能越高。

$$RL_D(f) = \frac{1}{n} \sum_{i=1}^n \frac{1}{|Y_i|} \left| \left\{ (y_1, y_2) \mid f(x_i, y_1) \leq f(x_i, y_2), (y_1, y_2) \in Y_i \times \bar{Y}_i \right\} \right| \quad (26)$$

排序损失是评估对象非属标记的排位高于所属标记的次数情况。当  $RL_D(f)=0$  时为最好情况, 即  $RL_D(f)$  越小,  $f$  的性能越高。

$$AP_D(f) = \frac{1}{n} \sum_{i=1}^n \frac{1}{|Y_i|} \sum_{y \in Y_i} \frac{\left| \left\{ y' \mid \text{rank}_f(x_i, y') \leq \text{rank}_f(x_i, y), y' \in Y_i \right\} \right|}{\text{rank}_f(x_i, y)} \quad (27)$$

平均精度是评估在特定标记  $y \in Y_i$  排列的正确标记的平均分数。当  $AP_D(f)=1$  时为最好情况, 即  $AP_D(f)$  越大,  $f$  的性能越高。

### 3.3 实验环境及实验方案

对比实验代码均在 Matlab2016a 中运行, 硬件环境 Intel® Core™ i5-7500 3.4 GHz CPU, 8 GB 内存; 操作系统为 Windows 10。为了验证算法的可靠性和有效性, 算法选择多标记的 5 种常用评价指标, 分别是: Hamming Loss、One Error、Coverage、Ranking Loss 和 Average Precision。通过评价指标来综合衡量各算法的性能, 评估各算法的性能。实验中将 5 种评价指标分别简写为: HL↓、OE↓、CV↓、RL↓和 AP↑。其中↑表示指标数值越高越好, ↓表示指标数值越低越好。对比实验算法采用 ML-KELM<sup>[21]</sup>、RELM<sup>[15]</sup>、ELM<sup>[18]</sup>3 种基于 ELM 的多标记算法, 以此来验证本文提出的 ML-

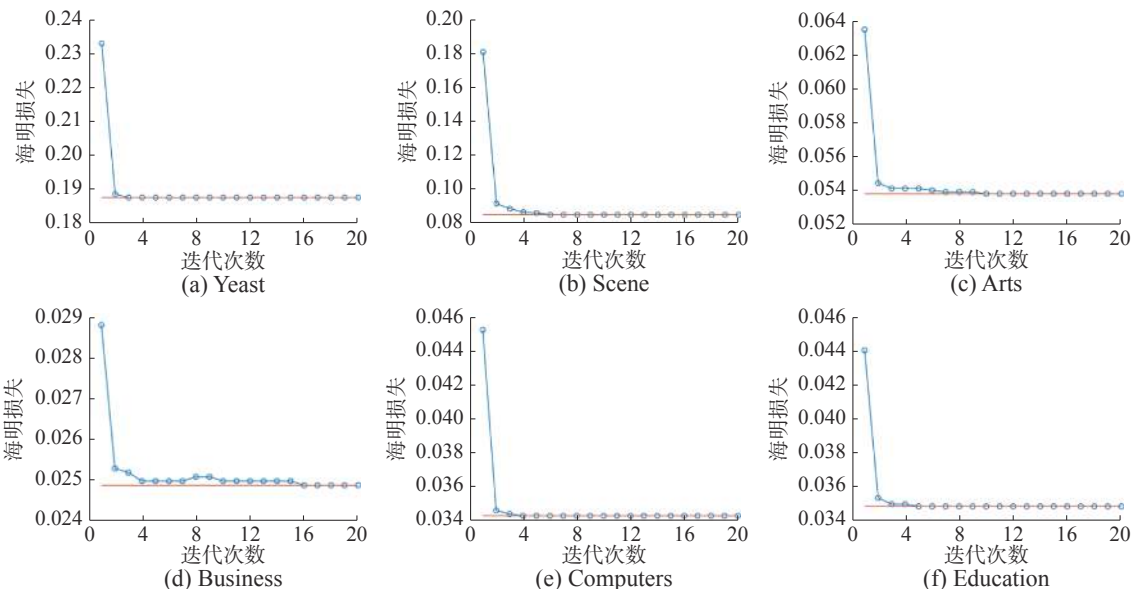
EKELM 算法较目前已提出基于 ELM 的多标记算法的优势, 同时对比 ML-RBF<sup>[13]</sup>、ML-KNN<sup>[10]</sup>2 种经典的多标记算法。

考虑算法对比验证的可行性和准确性, 减少随机误差的产生, 各测试算法在一个数据集中做 10 次实验, 最终将 10 次实验得到的 5 种评价指标求出平均值 (mean) 和标准差 (standard deviation)。在每个评价指标数据下标注排位情况, 如 ML-EKELM<sub>(1)</sub> 表示在某个数据集 ML-EKELM 算法最为优秀, 同时用黑体表示, 并在雅虎网页数据集给出了 11 个子集的平均评价指标数据 Average。

### 3.4 实验结果及分析

为了更直观展示本文算法收敛速度, 13 个数据集迭代收敛情况如图 2 所示。同时, 在 13 个数据集中对比实验结果如表 2~9 所示, 其中表 2 是酵母菌基因数据集对比试验结果, 表 3 为场景数据集对比实验实验结果, 表 4~8 则是雅虎网页数据集的实验结果, 表 9 给出各算法在 13 个数据集中实验的时间消耗, 并给出平均时间消耗。在此特别说明: 因算法 ML-EKELM、ML-KELM、ML-KNN 的分类器具有稳定性, 10 次实验结果相同, 其标准差均为 0。

图 2 为 ML-EKELM 算法在 13 个多标记数据集中以 Hamming Loss 为指标的迭代次数图, 最终收敛的 Hamming Loss 值用水平线表示。通过图 2 可以看出, 使用坐标下降法求解弹性网络正则的 ML-EKELM 算法, 迭代次数均小于 20 次, 在大部分数据集中都在 3 次左右迭代达到收敛, 在 Arts 和 Business 数据集中算法迭代收敛次数分别是 10 和 16 次。同时可以发现在 13 个数据集中只有 Business 数据集迭代收敛时出现波动, 这一



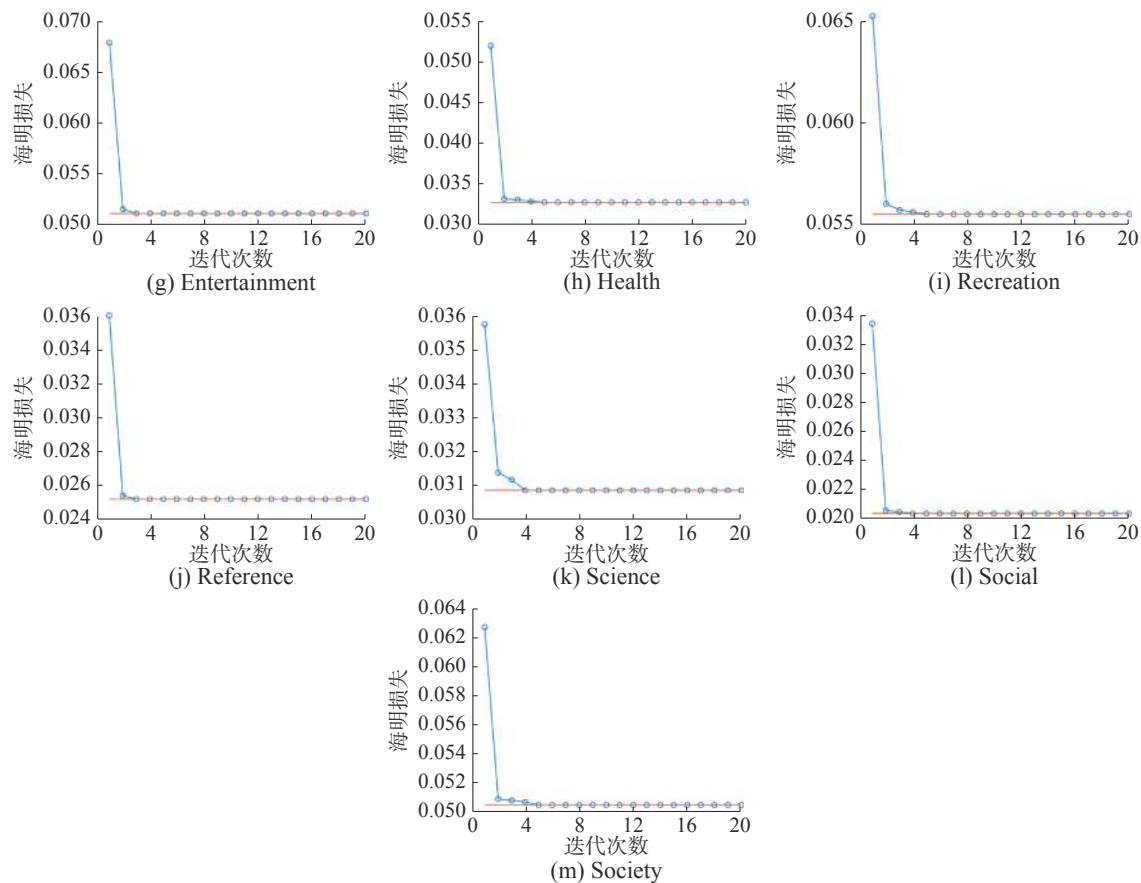


图2 ML-EKELM 迭代次数

Fig. 2 The number of ML-EKELM iterations

表2 酵母菌基因数据集测试结果

Table 2 Test results of Yeast Gene data set

算法	HL↓	OE↓	CV↓	RL↓	AP↑
ML-EKELM	<b>0.187 6±0<sub>(1)</sub></b>	<b>0.223 6±0<sub>(1)</sub></b>	<b>6.121 0±0<sub>(1)</sub></b>	<b>0.156 7±0<sub>(1)</sub></b>	<b>0.775 0±0<sub>(1)</sub></b>
ML-KELM	0.194 1±0 <sub>(2)</sub>	0.236 6±0 <sub>(4)</sub>	6.426 4±0 <sub>(3)</sub>	0.170 2±0 <sub>(2)</sub>	0.761 8±0 <sub>(2)</sub>
RELM	0.198 5±0.000 2 <sub>(5)</sub>	0.235 6±0.001 9 <sub>(3)</sub>	6.569 1±0.010 3 <sub>(5)</sub>	0.177 6±0.000 2 <sub>(5)</sub>	0.755 5±0.000 4 <sub>(5)</sub>
ELM	0.247 2±0.000 9 <sub>(6)</sub>	0.237 5±0.003 2 <sub>(6)</sub>	6.574 3±0.010 4 <sub>(6)</sub>	0.177 8±0.000 5 <sub>(6)</sub>	0.755 4±0.001 0 <sub>(6)</sub>
ML-RBF	0.197 8±0.001 3 <sub>(3)</sub>	0.237 2±0.006 9 <sub>(5)</sub>	6.496 3±0.028 4 <sub>(4)</sub>	0.173 6±0.002 3 <sub>(4)</sub>	0.758 6±0.002 3 <sub>(3)</sub>
ML-KNN	0.198 0±0 <sub>(4)</sub>	0.234 5±0 <sub>(2)</sub>	6.414 4±0 <sub>(2)</sub>	0.171 5±0 <sub>(3)</sub>	0.758 5±0 <sub>(4)</sub>

表3 场景数据集测试结果

Table 3 Test results of Scene data set

算法	HL↓	OE↓	CV↓	RL↓	AP↑
ML-EKELM	<b>0.086 0±0<sub>(1)</sub></b>	<b>0.204 8±0<sub>(1)</sub></b>	<b>0.469 9±0<sub>(1)</sub></b>	<b>0.073 2±0<sub>(1)</sub></b>	<b>0.875 5±0<sub>(1)</sub></b>
ML-KELM	0.092 0±0 <sub>(2)</sub>	0.235 8±0 <sub>(4)</sub>	0.483 3±0 <sub>(2)</sub>	0.076 7±0 <sub>(2)</sub>	0.860 4±0 <sub>(3)</sub>
RELM	0.128 4±0.001 5 <sub>(5)</sub>	0.235 0±0.009 2 <sub>(3)</sub>	0.553 0±0.012 0 <sub>(5)</sub>	0.088 3±0.002 4 <sub>(5)</sub>	0.854 8±0.005 0 <sub>(4)</sub>
ELM	0.134 6±0.001 3 <sub>(6)</sub>	0.241 5±0.007 1 <sub>(5)</sub>	0.549 0±0.016 6 <sub>(4)</sub>	0.088 0±0.003 0 <sub>(4)</sub>	0.852 5±0.003 9 <sub>(5)</sub>
ML-RBF	0.094 0±0.001 3 <sub>(3)</sub>	0.232 1±0.004 2 <sub>(2)</sub>	0.506 63±0.007 3 <sub>(3)</sub>	0.080 5±0.001 2 <sub>(3)</sub>	0.860 7±0.002 0 <sub>(2)</sub>
ML-KNN	0.098 9±0 <sub>(4)</sub>	0.242 5±0 <sub>(6)</sub>	0.568 6±0 <sub>(6)</sub>	0.093 1±0 <sub>(6)</sub>	0.851 2±0 <sub>(6)</sub>

表4 雅虎网页数据集海明损失测试结果

Table 4 Test results of hamming loss↓ on Yahoo Web Pages data set

数据集	ML-EKELM	ML-KELM	RELM	ELM	ML-RBF	ML-KNN
Arts	<b>0.0539±0<sub>(1)</sub></b>	0.054 3±0 <sub>(3)</sub>	0.054 5±0.000 1 <sub>(4)</sub>	0.058 6±0.000 1 <sub>(5)</sub>	0.054 2±0.000 1 <sub>(2)</sub>	0.059 3±0 <sub>(6)</sub>
Business	<b>0.024 9±0<sub>(1)</sub></b>	0.025 3±0 <sub>(2)</sub>	0.025 4±0.000 1 <sub>(3)</sub>	0.028 7±0.000 1 <sub>(6)</sub>	0.025 5±0.000 1 <sub>(4)</sub>	0.026 7±0 <sub>(5)</sub>
Computers	<b>0.034 4±0<sub>(1)</sub></b>	0.034 7±0 <sub>(3)</sub>	0.035 1±0.000 1 <sub>(4)</sub>	0.039 7±0.000 1 <sub>(6)</sub>	0.034 5±0.000 3 <sub>(2)</sub>	0.038 1±0 <sub>(5)</sub>
Education	<b>0.036 8±0<sub>(1)</sub></b>	0.037 6±0 <sub>(3)</sub>	0.037 7±0.000 1 <sub>(4)</sub>	0.040 9±0.000 1 <sub>(6)</sub>	0.037 5±0.000 2 <sub>(2)</sub>	0.039 7±0 <sub>(5)</sub>
Entertainment	<b>0.051 3±0<sub>(1)</sub></b>	0.053 0±0 <sub>(4)</sub>	0.052 5±0.000 1 <sub>(3)</sub>	0.058 2±0.000 1 <sub>(6)</sub>	0.051 5±0.000 4 <sub>(2)</sub>	0.057 0±0 <sub>(5)</sub>
Health	<b>0.032 9±0<sub>(1)</sub></b>	0.034 7±0 <sub>(2.5)</sub>	0.034 7±0.000 1 <sub>(2.5)</sub>	0.041 3±0.000 1 <sub>(6)</sub>	0.035 5±0.000 2 <sub>(4)</sub>	0.036 7±0 <sub>(5)</sub>
Recreation	0.055 6±0 <sub>(2)</sub>	0.056 1±0 <sub>(3)</sub>	0.056 5±0.000 1 <sub>(4)</sub>	0.059 7±0.000 1 <sub>(6)</sub>	<b>0.054 7±0.000 2<sub>(1)</sub></b>	0.059 4±0 <sub>(5)</sub>
Reference	<b>0.025 0±0<sub>(1)</sub></b>	0.025 7±0 <sub>(3.5)</sub>	0.025 7±0.000 1 <sub>(3.5)</sub>	0.029 3±0.000 1 <sub>(6)</sub>	0.025 2±0.000 1 <sub>(2)</sub>	0.027 3±0 <sub>(5)</sub>
Science	<b>0.030 1±0<sub>(1)</sub></b>	0.031 1±0 <sub>(3)</sub>	0.031 2±0.000 1 <sub>(4)</sub>	0.033 6±0.000 1 <sub>(6)</sub>	0.030 8±0.000 1 <sub>(2)</sub>	0.033 4±0 <sub>(5)</sub>
Social	<b>0.020 5±0<sub>(1.5)</sub></b>	<b>0.020 5±0<sub>(1.5)</sub></b>	0.020 6±0.000 1 <sub>(3.5)</sub>	0.024 4±0.000 1 <sub>(6)</sub>	0.020 6±0.000 1 <sub>(3.5)</sub>	0.021 9±0 <sub>(5)</sub>
Society	<b>0.050 6±0<sub>(1)</sub></b>	0.051 7±0 <sub>(3)</sub>	0.051 9±0.000 2 <sub>(4)</sub>	0.055 3±0.000 1 <sub>(6)</sub>	0.051 5±0.000 1 <sub>(2)</sub>	0.054 3±0 <sub>(5)</sub>
Average	<b>0.037 8±0.012 8<sub>(1)</sub></b>	0.038 6±0.013 0 <sub>(3)</sub>	0.038 7±0.013 0 <sub>(4)</sub>	0.042 7±0.013 2 <sub>(6)</sub>	0.038 1±0.012 7 <sub>(2)</sub>	0.041 3±0.014 0 <sub>(5)</sub>

表5 雅虎网页数据集1-错误率测试结果

Table 5 Test results of one-error↓ on Yahoo Web Pages data set

数据集	ML-EKELM	ML-KELM	RELM	ELM	ML-RBF	ML-KNN
Arts	<b>0.4583±0<sub>(1)</sub></b>	0.474 0±0 <sub>(2)</sub>	0.480 3±0.002 4 <sub>(4)</sub>	0.488 4±0.001 9 <sub>(5)</sub>	0.475 9±0.004 2 <sub>(3)</sub>	0.550 3±0 <sub>(6)</sub>
Business	<b>0.112 0±0<sub>(1)</sub></b>	0.116 3±0 <sub>(3)</sub>	0.116 8±0.000 9 <sub>(4)</sub>	0.117 5±0.001 0 <sub>(5)</sub>	0.114 6±0.001 2 <sub>(2)</sub>	0.121 0±0 <sub>(6)</sub>
Computers	<b>0.345 7±0<sub>(1)</sub></b>	0.357 7±0 <sub>(2)</sub>	0.361 0±0.003 2 <sub>(4)</sub>	0.366 4±0.003 1 <sub>(5)</sub>	0.358 5±0.004 3 <sub>(3)</sub>	0.414 3±0 <sub>(6)</sub>
Education	<b>0.453 7±0<sub>(1)</sub></b>	0.466 3±0 <sub>(2)</sub>	0.471 8±0.002 4 <sub>(3)</sub>	0.482 8±0.002 9 <sub>(5)</sub>	0.476 3±0.003 4 <sub>(4)</sub>	0.512 3±0 <sub>(6)</sub>
Entertainment	<b>0.399 7±0<sub>(1)</sub></b>	0.411 0±0 <sub>(3)</sub>	0.413 0±0.002 8 <sub>(4)</sub>	0.422 5±0.004 4 <sub>(5)</sub>	0.406 7±0.004 4 <sub>(2)</sub>	0.498 7±0 <sub>(6)</sub>
Health	<b>0.246 0±0<sub>(1)</sub></b>	0.263 7±0 <sub>(3)</sub>	0.271 6±0.003 4 <sub>(4)</sub>	0.275 8±0.004 7 <sub>(5)</sub>	0.260 2±0.003 3 <sub>(2)</sub>	0.304 0±0 <sub>(6)</sub>
Recreation	<b>0.459 7±0<sub>(1)</sub></b>	0.463 3±0 <sub>(2)</sub>	0.468 4±0.002 4 <sub>(3)</sub>	0.482 6±0.003 3 <sub>(5)</sub>	0.471 2±0.004 6 <sub>(4)</sub>	0.561 7±0 <sub>(6)</sub>
Reference	<b>0.357 3±0<sub>(1)</sub></b>	0.375 0±0 <sub>(3)</sub>	0.376 2±0.001 5 <sub>(4)</sub>	0.381 1±0.004 7 <sub>(5)</sub>	0.364 8±0.004 0 <sub>(2)</sub>	0.411 7±0 <sub>(6)</sub>
Science	<b>0.486 0±0<sub>(1)</sub></b>	0.494 3±0 <sub>(2)</sub>	0.499 2±0.003 3 <sub>(3)</sub>	0.511 3±0.003 9 <sub>(5)</sub>	0.506 1±0.003 6 <sub>(4)</sub>	0.557 7±0 <sub>(6)</sub>
Social	0.281 7±0 <sub>(2)</sub>	<b>0.280 7±0<sub>(1)</sub></b>	0.285 3±0.002 0 <sub>(4)</sub>	0.295 2±0.003 5 <sub>(5)</sub>	0.281 9±0.003 6 <sub>(3)</sub>	0.321 0±0 <sub>(6)</sub>
Society	<b>0.390 7±0<sub>(1)</sub></b>	0.397 0±0 <sub>(2)</sub>	0.400 7±0.001 7 <sub>(3)</sub>	0.408 2±0.003 7 <sub>(5)</sub>	0.404 5±0.001 5 <sub>(4)</sub>	0.428 0±0 <sub>(6)</sub>
Average	<b>0.362 8±0.112 9<sub>(1)</sub></b>	0.372 7±0.114 2 <sub>(2)</sub>	0.376 8±0.115 0 <sub>(4)</sub>	0.384 7±0.118 1 <sub>(5)</sub>	0.374 6±0.117 8 <sub>(3)</sub>	0.425 5±0.135 4 <sub>(6)</sub>

表6 雅虎网页数据集覆盖率测试结果

Table 6 Test results of coverage↓ on Yahoo Web Pages data set

数据集	ML-EKELM	ML-KELM	RELM	ELM	ML-RBF	ML-KNN
Arts	5.121 3±0 <sub>(2)</sub>	5.337 7±0 <sub>(3)</sub>	5.581 2±0.036 9 <sub>(5)</sub>	5.907 1±0.033 4 <sub>(6)</sub>	5.778 8±0.038 3 <sub>(4)</sub>	<b>4.805 0±0<sub>(1)</sub></b>
Business	2.502 3±0 <sub>(2)</sub>	2.511 7±0 <sub>(3)</sub>	2.543 0±0.029 4 <sub>(4)</sub>	2.808 3±0.025 7 <sub>(6)</sub>	2.773 3±0.037 4 <sub>(5)</sub>	<b>2.140 3±0<sub>(1)</sub></b>
Computers	<b>4.003 0±0<sub>(1)</sub></b>	4.706 7±0 <sub>(4)</sub>	4.608 3±0.033 9 <sub>(3)</sub>	4.940 9±0.061 7 <sub>(5)</sub>	5.063 4±0.083 8 <sub>(6)</sub>	4.023 3±0 <sub>(2)</sub>
Education	4.275 7±0 <sub>(2)</sub>	4.364 7±0 <sub>(3)</sub>	4.492 1±0.033 2 <sub>(4)</sub>	5.575 8±0.046 5 <sub>(6)</sub>	5.368 2±0.078 7 <sub>(5)</sub>	<b>3.403 7±0<sub>(1)</sub></b>
Entertainment	<b>3.001 3±0<sub>(1)</sub></b>	3.177 0±0 <sub>(3)</sub>	3.320 8±0.023 3 <sub>(4)</sub>	3.573 6±0.027 0 <sub>(5)</sub>	3.899 0±0.043 8 <sub>(6)</sub>	3.010 3±0 <sub>(2)</sub>
Health	3.293 3±0 <sub>(2)</sub>	3.378 7±0 <sub>(3)</sub>	3.598 3±0.045 7 <sub>(4)</sub>	4.035 5±0.045 9 <sub>(6)</sub>	3.774 8±0.041 0 <sub>(5)</sub>	<b>2.787 7±0<sub>(1)</sub></b>
Recreation	<b>4.094 3±0<sub>(1)</sub></b>	4.294 7±0 <sub>(4)</sub>	4.233 9±0.024 6 <sub>(3)</sub>	4.715 6±0.035 7 <sub>(6)</sub>	4.529 7±0.043 7 <sub>(5)</sub>	4.299 7±0 <sub>(2)</sub>
Reference	3.448 7±0 <sub>(2)</sub>	3.516 3±0 <sub>(3)</sub>	3.760 1±0.047 2 <sub>(4)</sub>	4.112 9±0.040 5 <sub>(6)</sub>	3.942 5±0.036 9 <sub>(5)</sub>	<b>2.769 7±0<sub>(1)</sub></b>
Science	6.270 3±0 <sub>(2)</sub>	6.557 3±0 <sub>(3)</sub>	6.984 9±0.070 7 <sub>(4)</sub>	7.558 2±0.062 0 <sub>(6)</sub>	7.282 8±0.058 5 <sub>(5)</sub>	<b>5.788 0±0<sub>(1)</sub></b>
Social	3.714 3±0 <sub>(2)</sub>	3.752 3±0 <sub>(3)</sub>	4.036 1±0.042 2 <sub>(4)</sub>	4.455 1±0.048 2 <sub>(6)</sub>	4.370 1±0.074 6 <sub>(5)</sub>	<b>2.928 0±0<sub>(1)</sub></b>
Society	5.763 0±0 <sub>(2)</sub>	6.029 3±0 <sub>(3)</sub>	6.273 1±0.028 9 <sub>(4)</sub>	6.617 9±0.050 3 <sub>(6)</sub>	6.334 1±0.042 4 <sub>(5)</sub>	<b>5.248 7±0<sub>(1)</sub></b>
Average	4.135 2±1.163 1 <sub>(2)</sub>	4.329 7±1.246 8 <sub>(3)</sub>	4.493 8±1.317 8 <sub>(4)</sub>	4.936 4±1.386 0 <sub>(6)</sub>	4.782 5±1.341 8 <sub>(5)</sub>	<b>3.745 9±1.170 9<sub>(1)</sub></b>



表7 雅虎网页数据集排序损失测试结果

Table 7 Test results of ranking loss↓ on Yahoo Web Pages data set

数据集	ML-EKELM	ML-KELM	RELM	ELM	ML-RBF	ML-KNN
Arts	<b>0.126 8±0<sub>(1)</sub></b>	0.1358±0 <sub>(3)</sub>	0.143 4±0.001 1 <sub>(4)</sub>	0.153 6±0.001 1 <sub>(6)</sub>	0.149 8±0.001 2 <sub>(5)</sub>	0.127 9±0 <sub>(2)</sub>
Business	0.041 2±0 <sub>(2)</sub>	0.041 5±0 <sub>(3)</sub>	0.042 2±0.000 6 <sub>(4)</sub>	0.047 2±0.000 5 <sub>(6)</sub>	0.046 3±0.000 6 <sub>(5)</sub>	<b>0.035 9±0<sub>(1)</sub></b>
Computers	<b>0.080 2±0<sub>(1)</sub></b>	0.096 7±0 <sub>(4)</sub>	0.094 9±0.000 8 <sub>(3)</sub>	0.102 8±0.001 3 <sub>(5)</sub>	0.105 8±0.002 0 <sub>(6)</sub>	0.081 2±0 <sub>(2)</sub>
Education	0.088 4±0 <sub>(2)</sub>	0.091 0±0 <sub>(3)</sub>	0.093 6±0.000 6 <sub>(4)</sub>	0.118 2±0.000 9 <sub>(6)</sub>	0.112 7±0.001 6 <sub>(5)</sub>	<b>0.077 0±0<sub>(1)</sub></b>
Entertainment	<b>0.100 3±0<sub>(1)</sub></b>	0.100 7±0 <sub>(2)</sub>	0.113 9±0.000 8 <sub>(4)</sub>	0.124 4±0.001 2 <sub>(5)</sub>	0.117 8±0.001 9 <sub>(6)</sub>	0.108 1±0 <sub>(3)</sub>
Health	<b>0.045 8±0<sub>(1)</sub></b>	0.051 7±0 <sub>(3)</sub>	0.056 3±0.000 8 <sub>(4)</sub>	0.065 6±0.000 9 <sub>(6)</sub>	0.058 7±0.000 7 <sub>(5)</sub>	0.047 3±0 <sub>(2)</sub>
Recreation	<b>0.138 2±0<sub>(1)</sub></b>	0.146 0±0 <sub>(3)</sub>	0.143 6±0.000 7 <sub>(2)</sub>	0.163 4±0.001 6 <sub>(6)</sub>	0.154 9±0.001 7 <sub>(4)</sub>	0.155 0±0 <sub>(5)</sub>
Reference	0.080 8±0 <sub>(2)</sub>	0.082 2±0 <sub>(3)</sub>	0.088 9±0.001 1 <sub>(4)</sub>	0.098 2±0.001 1 <sub>(6)</sub>	0.092 9±0.001 0 <sub>(5)</sub>	<b>0.068 9±0<sub>(1)</sub></b>
Science	0.114 6±0 <sub>(2)</sub>	0.120 8±0 <sub>(3)</sub>	0.130 0±0.000 7 <sub>(4)</sub>	0.142 6±0.001 4 <sub>(6)</sub>	0.136 2±0.001 3 <sub>(5)</sub>	<b>0.110 7±0<sub>(1)</sub></b>
Social	0.064 8±0 <sub>(2)</sub>	0.065 6±0 <sub>(3)</sub>	0.070 9±0.000 7 <sub>(4)</sub>	0.079 6±0.001 0 <sub>(6)</sub>	0.077 0±0.001 3 <sub>(5)</sub>	<b>0.053 7±0<sub>(1)</sub></b>
Society	0.133 4±0 <sub>(2)</sub>	0.141 0±0 <sub>(3)</sub>	0.148 1±0.000 6 <sub>(4)</sub>	0.158 5±0.001 4 <sub>(6)</sub>	0.149 1±0.001 1 <sub>(5)</sub>	<b>0.130 3±0<sub>(1)</sub></b>
Average	0.092 2±0.033 7 <sub>(2)</sub>	0.097 5±0.035 8 <sub>(3)</sub>	0.102 3±0.036 6 <sub>(4)</sub>	0.114 0±0.039 1 <sub>(6)</sub>	0.109 2±0.037 4 <sub>(5)</sub>	<b>0.090 5±0.038 5<sub>(1)</sub></b>

表8 雅虎网页数据集平均精度测试结果

Table 8 Test results of average precision ↑ on Yahoo Web Pages data set

数据集	ML-EKELM	ML-KELM	RELM	ELM	ML-RBF	ML-KNN
Arts	<b>0.627 3±0<sub>(1)</sub></b>	0.614 3±0 <sub>(2)</sub>	0.607 1±0.001 1 <sub>(3)</sub>	0.598 2±0.000 9 <sub>(5)</sub>	0.606 6±0.001 8 <sub>(4)</sub>	0.569 7±0 <sub>(6)</sub>
Business	<b>0.886 9±0<sub>(1)</sub></b>	0.882 9±0 <sub>(2)</sub>	0.882 3±0.000 4 <sub>(3)</sub>	0.877 4±0.000 8 <sub>(6)</sub>	0.880 5±0.001 2 <sub>(5)</sub>	0.882 2±0 <sub>(4)</sub>
Computers	<b>0.711 0±0<sub>(1)</sub></b>	0.700 9±0 <sub>(2)</sub>	0.699 5±0.001 3 <sub>(3)</sub>	0.692 2±0.001 8 <sub>(5)</sub>	0.696 0±0.002 6 <sub>(4)</sub>	0.662 0±0 <sub>(6)</sub>
Education	<b>0.646 0±0<sub>(1)</sub></b>	0.637 9±0 <sub>(2)</sub>	0.633 1±0.001 4 <sub>(3)</sub>	0.615 6±0.001 2 <sub>(5)</sub>	0.621 1±0.002 2 <sub>(4)</sub>	0.608 2±0 <sub>(6)</sub>
Entertainment	<b>0.690 4±0<sub>(1)</sub></b>	0.684 7±0 <sub>(2)</sub>	0.677 8±0.001 5 <sub>(4)</sub>	0.668 1±0.002 5 <sub>(5)</sub>	0.679 4±0.003 0 <sub>(3)</sub>	0.621 8±0 <sub>(6)</sub>
Health	<b>0.795 5±0<sub>(1)</sub></b>	0.782 4±0 <sub>(2)</sub>	0.777 5±0.001 7 <sub>(4)</sub>	0.768 6±0.002 5 <sub>(5)</sub>	0.781 2±0.001 2 <sub>(3)</sub>	0.756 2±0 <sub>(6)</sub>
Recreation	<b>0.634 2±0<sub>(1)</sub></b>	0.628 7±0 <sub>(2)</sub>	0.626 9±0.001 5 <sub>(3)</sub>	0.610 7±0.002 0 <sub>(5)</sub>	0.620 1±0.002 6 <sub>(4)</sub>	0.561 3±0 <sub>(6)</sub>
Reference	<b>0.720 2±0<sub>(1)</sub></b>	0.708 7±0 <sub>(3)</sub>	0.705 0±0.001 5 <sub>(4)</sub>	0.696 7±0.002 7 <sub>(5)</sub>	0.709 6±0.002 1 <sub>(2)</sub>	0.682 0±0 <sub>(6)</sub>
Science	<b>0.604 2±0<sub>(1)</sub></b>	0.596 8±0 <sub>(2)</sub>	0.588 0±0.002 1 <sub>(3)</sub>	0.574 4±0.002 2 <sub>(5)</sub>	0.583 0±0.002 0 <sub>(4)</sub>	0.548 9±0 <sub>(6)</sub>
Social	<b>0.773 5±0<sub>(1)</sub></b>	0.771 9±0 <sub>(2)</sub>	0.766 4±0.001 2 <sub>(3)</sub>	0.755 5±0.001 9 <sub>(5)</sub>	0.764 4±0.001 5 <sub>(4)</sub>	0.755 5±0 <sub>(6)</sub>
Society	<b>0.643 7±0<sub>(1)</sub></b>	0.636 4±0 <sub>(2)</sub>	0.630 4±0.001 2 <sub>(3)</sub>	0.621 3±0.002 1 <sub>(5)</sub>	0.627 4±0.001 2 <sub>(4)</sub>	0.618 1±0 <sub>(6)</sub>
Average	<b>0.703 0±0.086 5<sub>(1)</sub></b>	0.695 1±0.087 4 <sub>(2)</sub>	0.690 4±0.088 9 <sub>(3)</sub>	0.679 9±0.091 3 <sub>(5)</sub>	0.688 1±0.090 9 <sub>(4)</sub>	0.660 5±0.102 2 <sub>(6)</sub>

表9 时间测试结果

Table 9 The results of testing time

s

数据集	ML-EKELM	ML-KELM	RELM	ELM	ML-RBF	ML-KNN
Yeast Gene	0.319 5	0.222 6	0.267 2	0.263 2	0.183 7	0.585 6
Scene	0.396 3	0.120 5	0.192 6	0.187 9	0.525 3	1.845 8
Arts	1.487 5	0.739 9	0.790 8	0.776 1	1.308 6	4.853 0
Business	1.527 0	0.802 5	0.848 4	0.845 6	1.419 6	4.849 8
Computers	1.834 9	0.887 4	0.932 4	0.957 8	1.837 7	5.270 8
Education	1.845 6	0.868 4	0.914 3	0.893 3	1.607 5	5.277 2
Entertainment	1.107 4	0.685 9	0.710 9	0.700 9	1.356 8	3.040 7
Health	1.813 3	0.876 7	0.931 1	0.951 9	1.728 6	4.811 7
Recreation	1.405 0	0.694 8	0.740 9	0.739 7	1.321 1	4.854 3
Reference	1.463 4	0.894 0	0.919 7	0.900 4	2.327 8	4.806 1
Science	1.942 9	1.013 2	1.156 1	1.027 4	2.482 7	4.376 9
Social	1.957 1	1.046 2	1.134 8	1.075 6	5.334 9	5.388 8
Society	1.378 3	0.804 2	0.870 6	0.824 3	1.643 6	4.893 3
Average Time	1.421 4	0.742 8	0.800 8	0.780 3	1.775 2	4.219 5

波动表明 ML-EKELM 算法在收敛过程中遇到局部最小值并成功寻找到全局最小值,这也进一步说明该算法求解弹性网络具有较强的鲁棒性,并且效率较高。

表2中,在 Yeast Gene 数据集上与其他算法对比,ML-EKELM 算法在5种评价指标中均为第1,在 HL↓指标中较第2位算法降低3.5%损失;如表3所示,在 Scene 数据集中,本文 ML-EKELM 算法同样在5种评价指标中最为优秀,在 OE↓指标中比第2位算法降低11.8%错误率,同时在 AP↑指标中比第2位算法提高1.7%准确率;雅虎网页数据集包含11个子数据集,其中分别对每个评价指标在各个子数据集中做出比较,如表4所示,在 HL↓指标上,Arts、Business、Computers、Education、Entertainment、Health、Reference、Science 和 Society 等数据集中 ML-EKELM 性能最优,在 Recreation 数据集上,该算法位列第2,与第1位算法相差仅1.6%,在数据集 Social 上,ML-EKELM 与 ML-KELM 性能并列第1,在 HL↓的平均指标中可以看出,ML-EKELM 算法性能最优。在表5中,对比了不同算法在各个数据集上的 OE↓指标数值,其中在 Social 数据集上,ML-EKELM 较 ML-KELM 相差仅为0.3%,排位第2,在其他数据集中该指标均为最优;11个子数据集在 CV↓指标上如表6所示,ML-EKELM 算法在 Computers、Entertainment 和 Recreation 数据集中指标上最优,其他数据均为第2,与平均指标性能最优的 ML-KNN 算法相差10.4%。

在 RL↓指标上如表7所示,该算法在 Arts、Computers、Entertainment、Health 和 Recreation 等

数据集上,指标性能最优,在平均性能指标位列第2位,与平均指标性能最优的 ML-KNN 算法仅相差1.9%;在 AP↑指标上如表8所示,ML-EKELM 算法在各个数据集上的性能指标均为最优。在雅虎网页数据集中,可以看出在 CV↓和 RL↓评价指标上,ML-KNN 具有一定优势,但 HL↓、OE↓和 AP↑则排名靠后。而本文提出的算法在 HL↓、OE↓和 AP↑上都具有较大优势,在 CV↓和 RL↓上对比其他算法也处于优势地位。

各算法在多个数据集实验的时间消耗如表9所示,本文提出的算法 ML-EKELM 由于采用坐标下降法求解弹性网络,是一种迭代算法,所以在平均时间消耗上高于直接求解矩阵解析解的3种 ELM 算法77.5%~91.4%,但该算法平均时间消耗低于 ML-RBF 算法24.9%,平均时间消耗低于 ML-KNN 算法196.9%。从时间消耗可以看出 ML-EKELM 算法对比传统 ELM 算法有一定差距,但是对于其他多标记学习算法具有一定优势,ML-EKELM 兼具准确率高与时间消耗较低的特点。

为了更清晰地展示各算法在13个数据集上的相对性能,采用显著性水平为5%的 Nemenyi 检验<sup>[29]</sup>。当两个对比算法在各数据集中的平均排序差值小于或等于临界差(critical difference, CD),则认为这两个算法没有显著性差异;反之则2个算法有显著性差异。图3给出了在5种评价指标下各算法的性能,其 CD 值为2.0913,没有显著性差异的算法用实线相连,在图3评价指标子图中各算法坐标即平均排序位置,数值越小则算法性能越高。

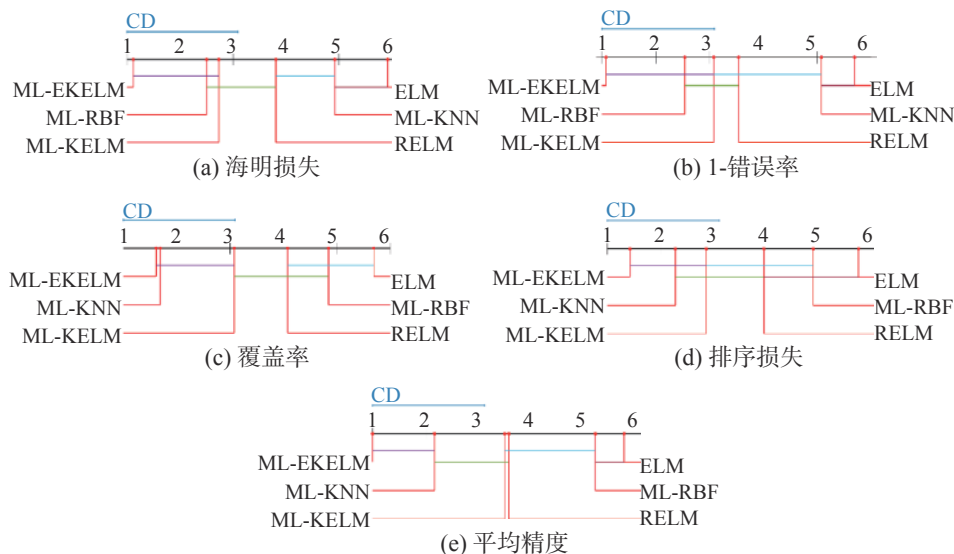


图3 算法性能比较

Fig. 3 The performance comparison of algorithms

对任意某个算法,都有25个结果作为对比(在5个评价指标上具有5个对比算法),通过图3可以得出:

1) 对于ML-EKELM算法,在5个评价指标上的性能均处于首位,除图3(c)中Coverage指标与第2位的ML-KNN相差不大,其余4个指标与第2位具有较大优势。在64%的情况下,统计上优于其它算法,如图3(a)在Hamming Loss指标上,ML-EKELM与RELM、ML-KNN和ELM有显著性差异,且优于这3种算法;如图3(b)在One-Error指标上,ML-EKELM与RELM、ELM和ML-KNN有显著性差异,且优于这3种算法;如图3(c)在Coverage指标上,ML-EKELM与RELM、ML-RBF和ELM有显著性差异,且优于这3种算法;如图3(d)在Ranking Loss指标上,ML-EKELM与RELM、ML-RBF和ELM有显著性差异,且优于这3种算法;如图3(e)在Average Precision指标上,ML-EKELM与RELM、ML-RBF、ELM和ML-KNN有显著性差异,且优于这4种算法。在36%情况下,与其它算法性能无显著性差异。

2) 对于ML-KELM算法,统计上优于其它对比算法有36%;与其它对比算法无显著性差异有64%。

3) 对于ML-KNN算法,有20%的情况,在统计上优于其它对比算法;有44%的情况,与其他对比算法无显著性差异;在36%的情况性能弱于其他算法。

通过以上对于图3的分析,ML-EKELM算法综合性能最为优秀,在统计上优于其他对比算法有64%;第2位的是ML-KELM算法,在36%的情况下,在统计上优于其它对比算法,第3则是ML-KNN算法,有20%的情况优于其他对比算法。

基于以上的实验结果和分析表明提出的基于弹性网络核极限学习机的多标记学习算法(ML-EKELM)在综合性能方面有较好的表现,是对于ELM解决多标记问题的一种补充。

## 4 结束语

本文首次提出基于弹性网络核极限学习机的多标记学习算法,通过弹性网络正则防止数据训练时过拟合情况发生,并可对核映射后特征进行稀疏化表示,即可对特征进行选择。对比传统使用岭回归正则化ELM算法,弹性网络正则式存在不可导点,所以采用非梯度优化的坐标下降法,而无需对目标函数求导。该算法对于多标记学习任务,在运行速度和分类精度上都具有一定优势,对比试验进一步说明算法的可靠性和稳定性。

但目前本文只将弹性网络和ELM结合运用到多标记学习中,对于弹性网络如何稀疏化特征空间以及进行特征选择并未深入研究和实验,这将是今后研究的一个重要方向和目标。

## 参考文献:

- [1] ZHANG Minling, ZHOU Zhihua. A review on multi-label learning algorithms[J]. *IEEE transactions on knowledge and data engineering*, 2014, 26(8): 1819–1837.
- [2] SCHAPIRE R E, SINGER Y. BoosTexter: a boosting-based system for text categorization[J]. *Machine learning*, 2000, 39(2/3): 135–168.
- [3] AGRAWAL S, AGRAWAL J, KAUR S, et al. A comparative study of fuzzy PSO and fuzzy SVD-based RBF neural network for multi-label classification[J]. *Neural computing and applications*, 2018, 29(1): 245–256.
- [4] BOUTELL M R, LUO Jiebo, SHEN Xipeng, et al. Learning multi-label scene classification[J]. *Pattern recognition*, 2004, 37(9): 1757–1771.
- [5] GUAN Renchu, WANG Xu, YANG M Q, et al. Multi-label deep learning for gene function annotation in cancer pathways[J]. *Scientific reports*, 2018, 8(1): 267.
- [6] TOMAR D, AGARWAL S. Multi-label classifier for emotion recognition from music[C]//Proceedings of the 3rd International Conference on Advanced Computing, Networking and Informatics. New Delhi, India: Springer, 2016: 111–123.
- [7] READ J, PFAHRINGER B, HOLMES G. Multi-label classification using ensembles of pruned sets[C]//Proceedings of the Eighth IEEE International Conference on Data Mining. Pisa: IEEE, 2008: 995–1000.
- [8] READ J. A pruned problem transformation method for multi-label classification[C]//Proceedings of 2008 New Zealand Computer Science Research Student Conference. Christchurch, New Zealand: NZCSRS, 2008: 143–150.
- [9] TSOU MAKAS G, VLAHAVAS I. Random  $k$ -labelsets: an ensemble method for multilabel classification[C]//Proceedings of the 18th European Conference on Machine Learning. Warsaw, Poland: Springer, 2007: 406–417.
- [10] ZHANG Minling, ZHOU Zhihua. ML-KNN: a lazy learning approach to multi-label learning[J]. *Pattern recognition*, 2007, 40(7): 2038–2048.
- [11] ZHANG Minling, PEÑA J M, ROBLES V. Feature selection for multi-label naive Bayes classification[J]. *Information sciences*, 2009, 179(19): 3218–3229.
- [12] ELISSEEFF A, WESTON J. A kernel method for multi-labelled classification[C]//Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic. Vancouver, Canada: MIT

- Press, 2001: 681-687.
- [13] ZHANG Minling.  $M_{L-RBF}$ : RBF neural networks for multi-label learning[J]. *Neural processing letters*, 2009, 29(2): 61-74.
- [14] HUANG Guangbin, ZHU Qinyu, SIEW C K. Extreme learning machine: theory and applications[J]. *Neurocomputing*, 2006, 70(1/2/3): 489-501.
- [15] 邓万宇, 郑庆华, 陈琳, 等. 神经网络极速学习方法研究[J]. *计算机学报*, 2010, 33(2): 279-287.
- DENG Wanyu, ZHENG Qinghua, CHEN Lin, et al. Research on extreme learning of neural networks[J]. *Chinese journal of computers*, 2010, 33(2): 279-287.
- [16] MICHE Y, VAN HEESWIJK M, BAS P, et al. TROP-ELM: a double-regularized ELM using LARS and Tikhonov regularization[J]. *Neurocomputing*, 2011, 74(16): 2413-2421.
- [17] HUANG Guangbin, ZHOU Hongming, DING Xiaojian, et al. Extreme learning machine for regression and multi-class classification[J]. *IEEE transactions on systems, man, and cybernetics, part B (cybernetics)*, 2012, 42(2): 513-529.
- [18] ER M J, VENKATESAN R, WANG Ning. A high speed multi-label classifier based on extreme learning machines[C]//*Proceedings of ELM-2015 Volume 2: Theory, Algorithms and Applications*. Cham: Springer International Publishing, 2016.
- [19] SUN Xia, XU Jingting, JIANG Changmeng, et al. Extreme learning machine for multi-label classification[J]. *Entropy*, 2016, 18(6): 225.
- [20] ZHANG Nan, DING Shifei, ZHANG Jian. Multi layer ELM-RBF for multi-label learning[J]. *Applied soft computing*, 2016, 43: 535-545.
- [21] LUO Fangfang, GUO Wenzhong, YU Yuanlong, et al. A multi-label classification algorithm based on kernel extreme learning machine[J]. *Neurocomputing*, 2017, 260: 313-320.
- [22] HAN Yahong, WU Fei, ZHUANG Yueting, et al. Multi-label transfer learning with sparse representation[J]. *IEEE transactions on circuits and systems for video technology*, 2010, 20(8): 1110-1121.
- [23] ZOU Hui, HASTIE T. Regularization and variable selection via the elastic net[J]. *Journal of the royal statistical society: series B (statistical methodology)*, 2005, 67(2): 301-320.
- [24] FRIEDMAN J, HASTIE T, TIBSHIRANI R. Regularization paths for generalized linear models via coordinate descent[J]. *Journal of statistical software*, 2010, 33(1): 1-22.
- [25] ZHOU Zhihua, ZHANG Minling. Multi-label learning[M]//SAMMUT C, WEBB G I. *Encyclopedia of Machine Learning and Data Mining*. Boston, MA: Springer, 2017: 875-881.
- [26] WRIGHT S J. Coordinate descent algorithms[J]. *Mathematical programming*, 2015, 151(1): 3-34.
- [27] DONOHO D L. De-noising by soft-thresholding[J]. *IEEE transactions on information theory*, 1995, 41(3): 613-627.
- [28] FRIEDMAN J, HASTIE T, HÖFLING H, et al. Pathwise coordinate optimization[J]. *The annals of applied statistics*, 2007, 1(2): 302-332.
- [29] DEMŠAR J. Statistical comparisons of classifiers over multiple data sets[J]. *Journal of machine learning research*, 2006, 7: 1-30.

#### 作者简介:



王一宾,男,1970年生,教授,主要研究方向为多标记学习、机器学习、软件安全。主持安徽省教育厅重点项目多项,发表学术论文20余篇。



裴根生,男,1992年生,硕士研究生,主要研究方向为机器学习、数据挖掘、统计。



程玉胜,男,1969年生,教授,博士,主要研究方向为数据挖掘、机器学习。主持省自然科学基金项目1项、省教育厅项目多项,发表学术论文50余篇。