

基于 LS-SVM 的多标签分类算法

殷 会, 许建华, 许 花

(南京师范大学 计算机科学与技术学院, 江苏 南京 210097)

[摘要] 多标签分类是指部分样本同时归属多个类别. 基于数据分解的算法因训练速度快、性能良好而得到广泛的应用. 本文采用一对一分解策略, 将  $k$  标签数据集分解为  $k(k-1)/2$  个两类单标签和两类双标签的数据子集. 对每一训练子集统一用 LS-SVM 模型建立子分类器, 当出现双标签样本时将其函数值设为 0 并确定适当的分类阈值. 对情感、景象和酵母数据集的实验结果表明, 本文算法的某些性能指标优于现有一些常用的多标签分类方法.

[关键词] LS-SVM, 多标签分类, 一对一分解

[中图分类号] TP391.4 [文献标识码] A [文章编号] 1672-1292(2010)02-0068-06

A Multi-Label Classification Algorithm Based on LS-SVM

Yin Hui, Xu Jianhua, Xu Hua

(School of Computer Science and Technology, Nanjing Normal University, Nanjing 210097, China)

**Abstract** A multi-label classification problem lies in that its samples may belong to multiple classes. Data decomposition algorithms are widely used because of its good performance. One versus one decomposition strategy is adopted in this paper, and this strategy decomposes a multi-label problem into several binary class single label or binary class double label classification sub-problems which can be solved independently. For each sub-problem, we build a sub-classifier using LS-SVM model and set the function value zero when the sample is double label, then determine a proper threshold. Experimental results show that our performance is superior to several existent multi-label classification algorithms with some evaluation criteria on three benchmark datasets Yeast, Scene and Emotion.

**Key words** LS-SVM, multi-label classification, one versus one decomposition strategy

分类是用训练样本建立的模型将测试样本分到一个或多个类中. 传统的单标签分类问题假设类间相互独立, 一个样本只能归为其中一类. 而在实际应用中, 样本会和多个类相关联, 需将样本同时归到多个类, 这就是多标签分类问题. 例如, 对于一幅侧重描绘山的山水画, 单标签分类就将其归为山类; 而多标签分类将它同时归为山类和水类, 能更准确和全面地反映出画的特性.

目前, 多标签分类算法的研究已经取得了很多成果, 大致可分为整体优化和基于数据分解两种方法. 整体优化方法对所有样本和标签构建一个优化问题, 如 Rank-SVM 算法<sup>[1]</sup>、Boostexter 算法<sup>[2]</sup>、多标签  $k$  近邻算法<sup>[3]</sup> (ML- $k$ NN) 以及最大化熵的多标签算法<sup>[4]</sup> (MIME) 等等. Rank-SVM 算法是多类 SVM 算法的推广, 其优化形式是一个二次规划问题, 在排序函数中选择一组标签子集作为样本的预测标签; Boostexter 算法根据弱分类器的分类结果, 提高错误分类样本的权重, 将弱分类器加权组合成强分类器的多标签分类方法; ML- $k$ NN 以  $k$  近邻算法为基础, 使用贝叶斯规则来计算样本的后验概率, 从而确定该样本是否属于某个标签; MIME 不仅考虑了样本与标签之间的关系, 还考虑了标签与标签之间的关系, 同时它还使用了一个正则化参数来调整经验风险和实际的分布, 从而避免产生过学习的情况. 整体优化方法的优点是没有改变数据的结构, 没有破坏类与类之间的联系, 但需要花费大量时间去解优化问题, 因此难以应用到较大规模的数据集. 为此, 很多学者转向更易于实现的基于数据分解的多标签分类算法.

基于数据分解的方法是将多标签数据集分解成数据子集, 如幂集法、一对多和一对一分解法等. 幂集

收稿日期: 2010-03-30  
基金项目: 国家自然科学基金 (60875001).  
通讯联系人: 许建华, 教授, 研究方向: 模式识别、机器学习和信号处理. E-mail: xujianhua@njnu.edu.cn

法<sup>[5]</sup>是将不同标签组合的样本作为新的单标签数据, 接着使用单标签的分类方法对新的单标签集进行分类. 由于幂集转化出一个多类问题, 为任何标签组合构成新的类别, 因此会产生大量样本很少的新类, 并在预测阶段不会产生任何新的标签组合, 所以该算法的应用并不多. 一对多分解<sup>[6]</sup>将  $k$  标签分类问题分解成  $k$  个两类单标签子问题. 一对一分解则分解成  $k(k-1)/2$  个两类单标签和两类双标签的分类子问题. 由于一对一分解出的数据集规模比一对多要小, 且分解出的两类样本的数量更平衡, 因此, 一对一的分解策略更受科研人员的青睐.

但是, 一对一分解策略会导致产生两类双标签的数据子集, 目前典型的处理方法有两次使用两类 SVM 的方法<sup>[7]</sup>和三类 SVM 算法<sup>[8]</sup>. 前者训练两次 SVM 以解决双标签问题; 后者将双标签作为独立类, 使用两个平行超平面分隔三类样本, 且需求得两个分类阈值. 上述两种算法都使用 SVM 方法对数据子集进行两类分类. 由于 SVM 是一个二次优化问题, 其算法复杂度为  $O(n^3)$ , 其中  $n$  代表样本点的数目, 这导致了其计算复杂度随着样本集合的增大迅速增加. 而 LS-SVM 建立的分类模型将 SVM 优化问题中不等式约束变为等式约束, 因此能够降低其复杂度. 获得 LS-SVM 模型仅需求解一个线性方程组, 其复杂度为  $O(n^2)$ , 其预测性能不亚于 SVM.

本文采用最小二乘 SVM 建立子分类器, 当出现两类双标签时, 将其标签值置为 0 并根据数据分布确定合适的分类阈值. 针对情感、景象和酵母数据集的实验结果说明, 本文的算法在某些性能指标上优于现有的方法.

## 1 一对一数据分解策略

设有  $k$  个标签的训练集  $\{x_i, y(x_i)\}_{i=1}^n$ , 其中  $x_i \in \mathbf{R}^n$ ,  $y(x_i) = [y_i^1, \dots, y_i^k] \in \{0, 1\}^k$ , 样本  $x_i$  属于类  $j$  当且仅当  $y_i^j = 1$ . 一对一分解策略将  $k$  标签分类问题分解成  $k(k-1)/2$  个两类单标签和两类双标签的分类子问题.

设  $M^{uv} (u = 1, \dots, k-1; v = u+1, \dots, k)$  为标签为  $u$  或  $v$  的样本子集, 若样本子集  $M^{uv}$  为两类单标签数据子集, 则直接将标签为  $u$  的样本函数值置为 +1, 标签为  $v$  的样本函数值置为 -1; 若样本子集  $M^{uv}$  为两类双标签数据子集, 对既属于标签  $u$  又属于标签  $v$  的样本函数值置为 0.

## 2 两类双标签 LS-SVM 模型

设训练子集  $M^{uv} = \{x_i^{uv}, z_i^{uv}\}_{i=1}^m$ ,  $z_i^{uv} \in \{-1, 0, 1\}$ , 其中训练子集样本函数值为 1、-1 和 0 的样本个数分别为  $n_1, n_2, n_3$  ( $n_1 + n_2 + n_3 = m$ ), 则基于  $M^{uv}$  建立 LS-SVM 模型即为求解式 (1) 的约束优化问题<sup>[9]</sup>.

$$\min \frac{1}{2} (\|w^{uv}\|_2)^2 + \frac{\gamma}{2} (\|\xi^{uv}\|_2)^2, \quad \text{s.t. } X^{uv} w^{uvT} + b^{uv} e + \xi^{uv} = z^{uv}, \quad (1)$$

式中,  $\|\cdot\|_2$  为 2 范数,  $w^{uv}$ ,  $b^{uv}$  分别为高维特征空间中线性判别函数的权向量和常数项,  $\gamma > 0$  为惩罚因子,  $X^{uv}$  为样本标签分别为 1、-1、0 的样本向量组成的矩阵,  $z^{uv}$  为  $n_1$  个 1、 $n_2$  个 -1、 $n_3$  个 0 组成的列向量,  $e$  为所有元素为 1 的列向量,  $\xi^{uv}$  为样本的松弛向量. 通过 Lagrange 乘子算法可将式 (1) 转换成其对偶形式, 并由最优性条件推导出式 (2) 所示的线性最小二乘问题:

$$\begin{bmatrix} K + \frac{1}{C} I \\ e^T \\ 0 \end{bmatrix} \begin{bmatrix} \alpha^{uv} \\ b^{uv} \end{bmatrix} = \begin{bmatrix} z^{uv} \\ 0 \end{bmatrix}, \quad (2)$$

式中,  $K$  为  $m \times m$  的核矩阵,  $I$  为单位阵,  $\alpha^{uv}$  为 Lagrange 乘子向量, 核矩阵  $K$  的元素为  $k_{ij} = k(x_i^{uv}, x_j^{uv})$ ,  $k(\cdot, \cdot)$  为核函数,  $x_i^{uv}, x_j^{uv} \in M^{uv}$  且  $i, j = 1, \dots, m$ , 本文选用 RBF 核, 表达式如 (3), 其中  $\sigma$  为 RBF 核的宽度.

$$k(x_i^{uv}, x_j^{uv}) = e^{-\frac{(\|x_i^{uv} - x_j^{uv}\|_2)^2}{\sigma^2}}. \quad (3)$$

求解式 (2) 可得到 LS-SVM 模型表达式 (4):

$$f^{uv}(x) = \sum_{i=1}^m \alpha_i^{uv} k(x, x_i^{uv}) + b^{uv}, \quad (4)$$

根据模型表达式  $f^w(x)$ , 判断分类结果, 如图 1 所示. 若  $f^w(x) \geq 0.5$  样本属于正类; 若  $f^w(x) \leq -0.5$  样本属于负类, 若样本  $-0.5 \leq f^w(x) \leq 0.5$  样本同时属于正类和负类.

3 两类分类阈值选择

建立两类双标签 LS-SVM 分类器时, 一般是将分类阈值  $t$  设为  $\pm 0.5$  这里的  $\pm 0.5$  是根据统计得来的. 由于阈值  $t$  与数据集分布有关, 为了得到更精确的预测结果, 本文根据训练样本的决策函数值确定阈值  $t$  的取值, 若训练样本不存在混合类, 直接将阈值  $t_p$  和  $t_n$  设为 0

由于负类和混合类的阈值求解与正类和混合类的情况相同, 本文仅解释正类和混合类之间的阈值是如何求得.

正类和混合类的阈值求解有以下两种情况:

- (1) 正类最小值大于等于混合类最大值, 即正类和混合类可以分开, 这种情况可以将间隙的平均值作为阈值, 即  $t_p = \frac{1}{2} [m \max(f_m(x)) + m \min(f_p(x))]$ , 其中  $f_m(x)$  和  $f_p(x)$  分别是混合类 and 正类函数值, 如图 2(a) 所示.
- (2) 正类最小值小于混合类最大值, 即两者有交叉, 这种情况需计算错分训练样本个数, 将错分最小时相邻两个样本点的平均值作为阈值  $t_p$ , 即  $t_p = \frac{1}{2} [s_{err-1}(x) + s_{err}(x)]$ , 其中  $s(x)$  为介于  $m \min(f_p(x))$  与  $m \max(f_m(x))$  之间排序过的样本函数值,  $err$  为错分最小的位置, 如图 2(b) 所示.

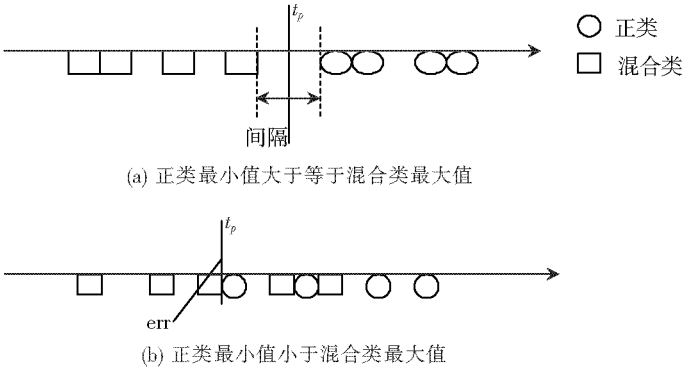


图 2 正类和混合类的阈值求解

Fig.2 Threshold solution by positive-mixing data distribution

根据上述不同的求阈值方法, 对景象数据集进行固定阈值和动态阈值的比较, 如表 1 所示. 从表 1 明显看出, 动态阈值的结果比固定阈值预测的结果要好, 由此可见, 优化阈值  $t$  能够提高算法预测结果的性能指标.

表 1 景象数据集固定阈值与设置动态阈值比较

Table 1 The comparison of fixed and dynamic threshold on scene datasets

景象	1 错误率 ↓	准确度 ↑	覆盖度 ↓	平均精度 ↑	精度 2↑	排序损失 ↓
固定阈值	0.299	0.665 2	0.941	0.707 4	0.724 2	0.167 5
动态阈值	0.229	0.737 3	0.857	0.820 4	0.765 8	0.102 3

4 多标签分类算法的框架

首先将训练样本根据一对一策略进行数据分解, 接着使用两类双标签 LS-SVM 建立分类器模型, 然后根据  $k(k-1)/2$  个预测结果, 将属于测试样本标签个数作为投票数, 投票函数记为  $V_j(x_i) (j = 1, \dots, k)$ , 根据  $V_j(x_i)$  和投票阈值  $T$  比较, 确定样本的标签. 其中, 多标签阈值的选择采用的方法是: 对训练集分别进

行训练和测试, 从 1 到  $2k$  中选择一个能使预测标签和真实标签相同的个数最多的值, 用该值作为投票阈值  $T$ .

算法具体描述如下:

训练阶段:

1. 用一对一策略将训练样本分解为  $k(k-1)/2$  个数据子集;
2. 建立 LS-SVM 模型, 得到  $k(k-1)/2$  个分类器;
3. 将训练样本经过每个 LS-SVM 子分类器, 得到决策函数值;
4. 对于双标签数据集, 根据训练样本的正类-混合类和负类-混合类函数分布计算  $t_p$  和  $t_n$ ;
5. 对训练集, 选择投票阈值  $T$ ;

测试阶段:

6. 将测试样本  $x_i$  经过标签为  $u, v (u = 1, \dots, k-1; v = u+1, \dots, k)$  的 LS-SVM 子分类器, 得到决策函数  $f^{uv}(x_i)$ ;
7. 根据决策函数  $f^{uv}(x_i)$  对  $x_i$  进行分类统计:
  - 7.1 若  $f^{uv}(x_i) \geq t_p$ , 则  $V_u(x_i)$  增 1;
  - 7.2 若  $f^{uv}(x_i) \leq t_n$ , 那么  $V_v(x_i)$  增 1;
  - 7.3 若  $t_n < f^{uv}(x_i) < t_p$ , 那么  $V_u(x_i)$  增 1,  $V_v(x_i)$  增 1;
8. 重复步骤 6, 直至样本  $x_i$  经过所有 LS-SVM 子分类器;
9. 若  $V_j(x_i) > T$ , 则  $y_i^j = 1$ , 否则  $y_i^j = 0$ .

## 5 多标签性能度量

为了测试多标签数据的分类结果, 需对多标签定义性能度量指标. 假设测试集为  $\{x_i, l(x_i)\}_{i=1}^p$ , 其中  $l(x_i) = [l_i^1, \dots, l_i^k] \in \{0, 1\}$ , 性能度量<sup>[13]</sup>有:

$$\text{汉明损失} = \frac{1}{p} \sum_{i=1}^p \frac{1}{k} \|y(x_i) \Delta l(x_i)\|_1, \quad (5)$$

式中,  $\Delta$  表示两个向量的对应分量做异或运算,  $\|\cdot\|_1$  表示向量的 1 范数.

$$1 \text{ 错误率} = \frac{1}{p} \sum_{i=1}^p |\{ \arg \max_{j=1, \dots, k} V_j(x_i) \} < T|, \quad (6)$$

$$\text{排序损失} = \frac{1}{p} \sum_{i=1}^p \frac{|\{ (r, s) \mid V_r(x_i) \leq V_s(x_i), \bar{l}_i = 1, \bar{l}_i = 1 \}|}{\|l(x_i)\|_1 \|\bar{l}(x_i)\|_1}, \quad (7)$$

式中,  $\bar{l}(x_i)$  为  $l(x_i)$  中对应分量做非运算,  $|\cdot|$  表示集合中元素的个数.

$$\text{平均精确度} = \frac{1}{p} \sum_{i=1}^p \frac{1}{\|l(x_i)\|_1} \sum_{h=1}^{\|l(x_i)\|_1} \left[ \frac{|\{ r \mid V_r(x_i) \geq V_y(x_i), \bar{l}_i = 1 \}|}{|\{ s \mid V_s(x_i) \geq V_y(x_i), s \in \{1, \dots, k\} \}|} \right], \quad (8)$$

$$\text{准确度} = \frac{1}{p} \sum_{i=1}^p \frac{\|y(x_i) \wedge l(x_i)\|_1}{\|y(x_i) \vee l(x_i)\|_1}, \quad (9)$$

式中,  $\wedge$  表示两个向量对应分量做与运算;  $\vee$  表示两个向量对应分量做或运算.

$$\text{精确度} = \frac{1}{p} \sum_{i=1}^p \frac{\|y(x_i) \wedge l(x_i)\|_1}{\|y(x_i)\|_1}, \quad (10)$$

$$\text{召回率} = \frac{1}{p} \sum_{i=1}^p \frac{\|y(x_i) \wedge l(x_i)\|_1}{\|l(x_i)\|_1}. \quad (11)$$

## 6 实验结果及分析

本文采用情感、景象和酵母数据集, 其具体的信息包括训练样本数、测试样本数、样本特征数、标签数及平均标签长度, 如表 2 所示. 酵母数据集<sup>[10]</sup>是一个关于基因功能分类的数据集, 其中每个样本代表一个基因, 它的特征来自于基因的微阵列表示和系统发育谱; 景象数据集<sup>[11]</sup>的每个样本代表一幅图像, 样本的特征取自于图像的颜色信息和结构信息; 情感数据集<sup>[5]</sup>的每个样本代表人们听到某种音乐所产生的情感, 样本的特征取自于音乐的节奏和音色.

表 2 基准数据集  
Table 2 Benchmark datasets

数据集	样本		特征数	标签数	平均标签长度
	训练	测试			
情感数据	391	202	72	6	1.87
景象数据	1 211	1 196	294	6	1.08
酵母数据	1 500	917	103	14	4.25

实验主要是比较本文所采用的基于 LS-SVM 双标签算法与一对多 SVM 算法、一对一 SVM 算法等的预测性能。一对一 SVM 算法采用一对一分解策略,使用两个两类 SVM 处理分解后的子问题,最后使用投票策略将子问题的结果进行预测;一对多 SVM 算法采用一对多分解策略,使用两类 SVM 来处理分解后的子问题;双标签 SVM 方法是将同时拥有正类标签和负类标签的样本看成双标签样本,使用样本复制的方法,将其标签同时标记为 + 1 和 - 1。其中一对一 SVM 算法和一对多 SVM 算法中使用的两类 SVM 算法均来自 Joachims 的 SVM<sup>light</sup>。此外,本文还引用了其他多标签分类算法的实验结果,包括一对多 C4.5、一对多  $k$  近邻、一对多朴素贝叶斯、一对多 SMO 等,这些方法引用于文献 [13-14]。

对于采用 RBF 核函数的基于 SVM 多标签分类模型需要找两个参数  $\gamma$  和  $\sigma^2$ ,对于一般的基于 SVM 的多标签分类算法,这两个参数都是通过人工寻找,来使预测结果达到最优;而 LS-SVM 模型则采用网格搜索的方法,设定这两个参数的可行区间,由计算机自动对各参数变量组合并逐一择优,使用留一法选取最佳的参数值,这种参数的估值方法可以保证所求得分类器是最优的。

情感、景象和酵母数据集中两个参数  $\gamma$  和  $\sigma^2$  的取值范围分别为  $10^{-2}$  至  $10^2$ 、 $10^{-3}$  至  $10^5$ 、 $10^{-2}$  至  $10$  之间。通过实验,确定情感数据集的投票阈值  $T$  为 5,景象数据集的投票阈值为 5,酵母数据集的投票阈值为 13。表 3~表 5 中的三类 SVM、一对一 SVM、一对多 SVM 以及双标签 SVM 等的结果均引用于文献 [14]。

表 3 情感数据比较  
Table 3 The comparison on emotion datasets

方法	汉明损失↓	准确度↑	查全率↑	平均精度↑	1 错误率↓	排序损失↓
三类 SVM	0.267	0.545	0.753	0.782	0.390	0.288
一对一 SVM	0.228	0.545	0.632	0.790	0.365	0.286
一对多 SVM	0.213	0.386	0.521	0.715	0.435	0.401
双标签 SVM	0.219	0.547	0.669	0.789	0.370	0.285
LS-SVM 多标签分类	0.205	0.565	0.700	0.722	0.292	0.213

表 4 景象数据比较  
Table 4 The comparison on scene datasets

方法	汉明损失↓	准确度↑	查全率↑	精确度 2↑
一对多 $k$ 近邻	0.125	0.637	0.669	0.651
一对多 C4.5	0.139	0.513	0.534	0.611
一对多朴素贝叶斯	0.247	0.435	0.443	0.816
一对多 SMO	0.114	0.571	0.596	0.628
一对一 SVM	0.084	0.754	0.770	0.787
一对多 SVM	0.091	0.689	0.709	0.716
三类 SVM	0.089	0.746	0.800	0.775
双标签 SVM	0.098	0.701	0.853	0.716
LS-SVM 多标签分类	0.081	0.665	0.878	0.724

由表 3、表 4 和表 5 看出,情感数据的比较结果表明,本文的方法在汉明损失、准确度、1 错误率以及排序损失上都有较好的结果,比其他方法的性能有较大的提高,而剩下的几个评价标准也均列在前列;而景象数据的比较结果表明,本文采用的方法在汉明损失和查全率上有较好的结果;从酵母数据实验的比较可以看出,本文采用的方法在查全率上比其他方法要好,但是汉明损失和准确度比双标签 SVM 稍差。

表 5 酵母数据比较  
Table 5 The comparison on yeast datasets

方法	汉明损失↓	准确度↑	查全率↑	精确度↑
一对多 $k$ 近邻	0.243	0.479	0.601	0.596
一对多 C4.5	0.259	0.423	0.593	0.561
一对多朴素贝叶斯	0.301	0.421	0.531	0.610
一对多 SMO	0.200	0.502	0.711	0.579
一对一 SVM	0.199	0.565	0.718	0.670
一对多 SVM	0.187	0.521	0.583	0.742
三类 SVM	0.196	0.538	0.652	0.697
双标签 SVM	0.181	0.572	0.697	0.606
LS-SVM 多标签分类	0.205	0.560	0.728	0.666

7 结语

本文使用一对一分解策略, 将多标签问题分解为  $k(k-1)/2$  个两类单标签和两类双标签的子数据集, 接着对其分解的数据集建立两类双标签 LS-SVM 分类器模型, 然后对两类分类的阈值进行优化, 最后利用投票方法将测试样本分到一个或多个类中. 对 4 个数据集的实现结果表明, 本文的算法在某些性能指标上优于现有一些常用的多标签分类算法.

[参考文献] (References)

[1] Elisseeff A, Weston J. A kernel method for multi-labelled classification[C] // Proceedings of Advances in Neural Information New York: BIOwolf Technologies, 2003: 681-687.

[2] Schapire R E, Singer Y. Boostexter: a boosting based system for text categorization[J]. Machine Learning, 2000, 39(2/3): 135-168.

[3] Zhang M L, Zhou Z H. A  $k$ -nearest neighbor based algorithm for multi-label classification[C] // Proceedings of the IEEE International Conference on Granular Computing Heidelberg: Springer Berlin, 2004: 718-721.

[4] Zhu S H, Ji X, Xu W, et al. Multi-labelled classification using maximum entropy method[C] // Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development. Salvador: ACM, 2004: 274-281.

[5] Trohidis K, Tsoumakas G, Kalliris G, et al. Multilabel classification of music into emotions[C] // Proceedings International Conference on Music Information Retrieval Philadelphia: ISMIR, 2008: 325-330.

[6] Tsoumakas G, Katakis I. Multi-label classification: an overview[J]. International Journal of Data Warehousing and Mining, 2007, 3(3): 1-13.

[7] Li T, Zhang C L, Zhu S H. Empirical studies on multi-label classification[C] // Proceedings of IEEE International Conference on Tools with Artificial Intelligence. Washington DC: IEEE Computer Society, 2006: 86-92.

[8] Wan S P, Xu J H. A multi-label classification algorithm based on triple class support vector machine[C] // Proceedings of IEEE International Conference on Wavelet Analysis and Pattern Recognition. Beijing: IEEE ICWAPR, 2007: 1447-1452.

[9] Suykens J K. Least squares support vector machines for classification and nonlinear modeling[J]. Neural Network World, 2000, 10: 29-48.

[10] Pavlidis P, Weston J, Cai J, et al. Combining microarray expression data and phylogenetic profiles to learn functional categories using support vector machines[C] // Proceedings of Annual International Conference on Computational Molecular Biology. Columbia: Columbia University, 2001: 242-248.

[11] Boult M R, Luo J, Shen X, et al. Learning multi-label scene classification[J]. Pattern Recognition, 2003, 37(9): 1757-1771.

[12] Tsoumakas G, Katakis I. Multi-label classification: an overview[J]. International Journal of Data Warehousing and Mining, 2007, 3(3): 1-13.

[13] Tsoumakas G, Katakis I, Vlahavas I. Mining Multi-label Data Data Mining and Knowledge Discovery Handbook[M]. New York: Springer, 2010.

[14] 李佳阳. 基于双标签支持向量机的快速多标签分类算法[D]. 南京: 南京师范大学计算机科学与技术学院, 2010.  
Li Jiayang. A fast multi-label classification algorithm based on double label support vector machine[D]. Nanjing: School of Computer Science and Technology, Nanjing Normal University, 2010. (in Chinese)

[责任编辑: 严海琳]