

# 基于关联规则和拓扑序列的分类器链方法<sup>\*</sup>

丁家满<sup>1,2</sup>, 周蜀杰<sup>1,2</sup>, 李润鑫<sup>1,2</sup>, 付晓东<sup>1,2</sup>, 贾连印<sup>1,2</sup>



<sup>1</sup>(昆明理工大学 信息工程与自动化学院, 云南 昆明 650500)

<sup>2</sup>(云南省人工智能重点实验室(昆明理工大学), 云南 昆明 650500)

通信作者: 贾连印, E-mail: [JLianyin@163.com](mailto:JLianyin@163.com)

**摘要:** 在分类器链方法中, 如何确定标签学习次序至关重要, 为此, 提出一种基于关联规则和拓扑序列的分类器链方法 (TSECC). 首先结合频繁模式设计了一种基于强关联规则的标签依赖度量策略; 接下来通过标签间依赖关系构建有向无环图, 对图中所有顶点进行拓扑排序; 最后将得到的拓扑序列作为分类器链方法中标签的学习次序, 对每个标签的分类器依次迭代更新. 特别地, 为减少无标签依赖或标签依赖度较低的“孤独”标签对其余标签预测性能的影响, 将“孤独”标签排在拓扑序列之外, 利用二元关联模型训练. 在多种公共多标签数据集上的实验结果表明, TSECC 能够有效提升分类性能.

**关键词:** 多标签学习; 分类器链; 标签依赖; 强关联规则; 拓扑序列; 二元关联

**中图法分类号:** TP18

中文引用格式: 丁家满, 周蜀杰, 李润鑫, 付晓东, 贾连印. 基于关联规则和拓扑序列的分类器链方法. 软件学报, 2023, 34(9): 4210-4224. <http://www.jos.org.cn/1000-9825/6659.htm>

英文引用格式: Ding JM, Zhou SJ, Li RX, Fu XD, Jia LY. Classifier Chains Method Based on Association Rules and Topological Sequences. Ruan Jian Xue Bao/Journal of Software, 2023, 34(9): 4210-4224 (in Chinese). <http://www.jos.org.cn/1000-9825/6659.htm>

## Classifier Chains Method Based on Association Rules and Topological Sequences

DING Jia-Man<sup>1,2</sup>, ZHOU Shu-Jie<sup>1,2</sup>, LI Run-Xin<sup>1,2</sup>, FU Xiao-Dong<sup>1,2</sup>, JIA Lian-Yin<sup>1,2</sup>

<sup>1</sup>(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China)

<sup>2</sup>(Artificial Intelligence Key Laboratory of Yunnan Province (Kunming University of Science and Technology), Kunming 650500, China)

**Abstract:** The order of label learning is crucial to a classifier chains method. Therefore, this study proposes a classifier chains method based on the association rules and topological sequence (TSECC). Specifically, a measurement strategy for label dependencies based on strong association rules is designed by leveraging frequent patterns. Then, a directed acyclic graph is constructed according to the dependency relationships among the labels to topologically sort all the vertices in the graph. Finally, the topological sequence obtained is used as the order of label learning to iteratively update each label's classifier successively. In particular, to reduce the impact of “lonely” labels with no or low label dependencies on the prediction performance on the other labels, TSECC excludes “lonely” labels out of the topological sequence and uses a binary relevance model to train them separately. Experimental results on a variety of public multi-label datasets show that TSECC can effectively improve classification performance.

**Key words:** multi-label learning; classifier chains; label dependent; strong association rules; topological sequence; binary relevance

传统监督学习通常假设每个实例与一个标签相关联, 而在现实世界的应用中, 一个实例通常有多个标签. 例如, 一篇关于奥运会的文档可能同时与“体育”“商业”和“经济”相关联. 传统的基于每个实例一个标签的监督学习已经无法解决这一问题, 而多标签学习处理与一组标签相关联的实例, 现已成为机器学习和数据挖掘领域中的研究热点, 并广泛应用于文本分类、图像标注和功能基因预测等领域中<sup>[1,2]</sup>.

\* 基金项目: 国家自然科学基金 (61562054)

收稿时间: 2021-09-23; 修改时间: 2021-11-29, 2022-01-08, 2022-02-16; 采用时间: 2022-02-24; jos 在线出版时间: 2022-12-22

CNKI 网络首发时间: 2022-12-26

处理多标签学习问题的常见策略是通过构造  $n$  个独立的二元分类器, 将其转换成单标签分类问题. 如二元关联法 (binary relevance, BR), 但这种方式忽略了标签间的隐含关系<sup>[3]</sup>. 合理利用标签间的隐含关系可以提高多标签分类的性能, 这也是近年来多标签学习研究的热点之一. 为此, Read 等人<sup>[4,5]</sup>提出分类器链方法 (classifier chains), 将标签按照某种次序排成链, 针对链上的标签依次训练分类器, 并将目标标签前的所有标签学习结果添加到目标标签分类器的训练特征中. 分类器链方法简单易实现, 且利用标签间的隐含关系取得了相对二元关联法更好的预测性能. 然而, 分类器链方法在学习过程中需预先给定标签的学习次序, 而良好的标签学习次序往往难以确定. 此外, 不当的标签学习次序会影响方法的预测性能, 甚至可能传递错误的信息<sup>[1]</sup>. Read 等人<sup>[4,5]</sup>利用集成思想, 提出了改进的集成分类器链方法 (ensemble classifier chains), 它随机产生多个的不同标签学习次序, 对多个次序的学习结果进行集成, 削弱了由单个标签序列产生的随机性, 在一定程度上缓解了分类器链方法性能受限的问题. 然而, 随着标签数量上升, 标签次序的排列组合数量呈阶乘级增长, 这极大地增加了集成分类器链方法预测的随机性和不确定性, 集成分类器链方法仍面临着次序选择的困难<sup>[6]</sup>.

针对上述问题, 本文提出基于关联规则和拓扑序列的分类器链方法, 方法首先结合频繁模式思想设计了一种基于强关联规则的标签依赖度量策略, 接下来通过标签间依赖关系构建有向无环图, 对其进行拓扑排序得到蕴含标签间依赖关系的拓扑序列, 最后将得到的拓扑序列作为分类器链方法中标签的学习次序, 对每个标签的分类器依次迭代更新. 在多种公共多标签数据集上的实验结果表明, TSECC 能够有效提升分类性能.

本文第 1 节介绍了多标签学习的相关工作. 第 2 节给出了本文方法的原理和实现流程. 第 3 节汇报实验结果及对比分析. 第 4 节总结全文.

## 1 相关工作

分类器链 (classifier chains) 算法<sup>[4,5]</sup>是一种由二元关联 (binary relevance) 算法<sup>[3]</sup>发展而来的可解释性强、易实现的多标签分类算法. 二元关联算法将多标签问题转换为多个独立的二分类问题, 为每个标签分别训练独立的二元分类器, 但这种算法忽略了标签关联信息, 性能受限. 分类器链算法在其基础上, 把已训练的标签加入到当前待训练标签分类器的特征中, 将多个相互独立的二元分类器串联起来, 使当前标签学习到前若干个标签信息, 提升了分类器的性能, 是一种利用标签关联信息的“高阶策略”的多标签学习方法<sup>[1]</sup>. 在多标签学习中, 合理地利用标签间隐含关系可以带来更加理想的预测性能, 这也是近年来多标签学习的研究重点. 例如, Wang 等人结合深度信念网络和反向传播神经网络传递标签间信息<sup>[7]</sup>; Dai 等人提出了基于模糊互信息的标签间对称不确定性关联<sup>[8]</sup>; Lin 等人构建了一个投影矩阵模型来确定多个标签间的关系<sup>[9]</sup>; Xie 等人通过标签间的条件熵构建标签间的关联关系<sup>[10]</sup>; Sun 等人提出了多重树增强的分类器链模型, 利用估计的条件互信息构建标签的多重树结构, 通过灵活的多重树结构来建模标签间的关联信息<sup>[11]</sup>. 从利用标签关联信息的角度, 多标签学习方法可大致分成以下 3 类<sup>[1]</sup>.

一阶方法, 即考虑标签间相互独立, 忽略标签的共存性. 例如, Zhang 等人提出的 ML-KNN 算法<sup>[12]</sup>, 通过计算样本  $K$  近邻, 利用最大后验概率原则预测样本的标签集合, 将 KNN 算法适应于多标签分类问题. 这种方式概念简单, 效率高, 但忽略了标签间的隐含关系, 学习性能受限.

二阶方法, 即学习标签间的成对关联关系, 例如, Elisseeff 等人提出的 Rank-SVM 算法<sup>[13]</sup>, Ghamrawi 等人提出的 CML 算法等<sup>[14]</sup>, Zhang 等人提出的一种三路选择集成模型<sup>[15]</sup>. 二阶方法简单有效, 且获得了良好的泛化性能, 但实际应用中标签间的隐含关系往往超出了二阶.

高阶方法, 即学习多个标签间的隐含关系. 高阶方法更准确地刻画了标签间的隐含关系, 具有相对更强的关联建模能力. 例如, Huang 等人提出的一种新的成本敏感型 LE 算法<sup>[16]</sup>. 此外, 分类器链方法是一种属于高阶方法的简单有效的算法.

对比二元关联算法, 分类器链算法将独立的二元分类器由并行结构改为了串联结构, 利用了标签关联信息, 提升了模型性能, 但其链式的串行结构使得模型对标签学习次序敏感, 不同的标签学习次序决定了不同的标签关联信息的学习程度, 因此分类器链算法性能不稳定, 具有很强的随机性, 大量实验结果表明, 学习次序的选择会严重影响分类器链算法的学习性能<sup>[6,17]</sup>, 针对此问题, Read 等人<sup>[4,5]</sup>提出了集成分类器链算法, 该方法构造多条标签学

习次序不同的分类器链, 每条分类器链的标签学习次序随机产生, 最终预测结果由多条分类器链集成得到. 集成分类器链算法通过集成多条不同标签学习次序的分类器链以削弱单条链带来的随机性. 但标签次序的排列组合数量随标签数量呈阶乘级增长, 在面对高纬度数据集时仍具有较强的随机性. 为此, 王少博等人<sup>[18]</sup>提出分类器圈算法, 将分类器链结构首尾链接, 使用圈结构的模型避免学习次序选择的问题; 李娜等人<sup>[19]</sup>提出基于多标签重要性排序的分类器链算法, 将标签间相互作用程度的大小作为衡量标签重要程度的依据, 用重要度排序结果作为分类器链模型的学习次序; Wang 等人<sup>[20]</sup>通过特征选择消除不相关特征的干扰, 并在圈结构训练的基础上, 将高耦合的标签加入到当前标签的训练特征中, 提升标签关联信息的表达; 胡天磊等人<sup>[21]</sup>提出双向分类器链的解决方案, 通过引入逆向链, 使标签再次提取不同的标签关联信息; Weng 等人<sup>[22]</sup>提出基于标签特征选择的分类器链方法, 通过对每个标签进行特征选择, 降低标签关联信息传递中冗余特征带来的干扰. 此外, Liu 等人<sup>[23]</sup>考虑到不平衡问题, 提出了基于随机欠采样的分类器链方法. Sun 等人<sup>[24]</sup>从条件似然最大化的角度对分类器链性能进行优化. 针对分类器链算法对标签学习次序敏感的问题, 分类器链相关改进算法多是通过多次迭代训练或改变模型结构克服标签学习次序问题的, 仍具有一定的随机性.

为了克服以上困难, 本文提出一种基于关联规则和拓扑序列的分类器链方法.

## 2 基于关联规则和拓扑序列的分类器链模型

### 2.1 算法思想

方法首先挖掘多标签数据集中的标签间依赖信息, 由依赖程度和依赖关系构建标签间依赖有向无环图, 接着对其进行拓扑排序, 求得标签次序, 最后将标签次序用于改进的分类器链算法. 如图 1 为本文方法的流程图.

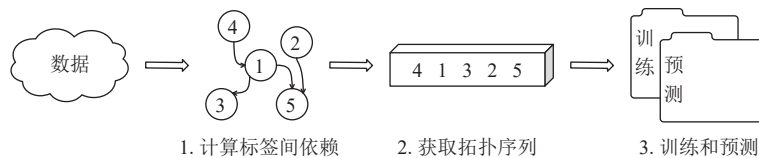


图 1 本文基于关联规则和拓扑序列的分类器链方法的流程图

方法分 3 个步骤, 首先是标签间依赖关系及依赖程度的挖掘. 本文提出一种计算标签间依赖的新方法, 迁移频繁模式强关联规则的思想, 将数据集中的一组组多标签向量类比为一个个事务, 探究标签项集间的关联规则, 将其作为标签间依赖关系及依赖程度.

接着是标签间依赖有向无环图的构建和其拓扑序列的求解. 将标签间依赖关系看作标签间约束条件, 以此构建有向无环图表示标签间依赖关系, 从而将标签关联信息隐含在图的拓扑序列中.

最后将求得的拓扑序列作为标签学习次序, 按学习次序训练分类器链模型, 并利用二元关联模型训练无标签依赖或标签依赖度较低的标签, 提高模型的泛化能力.

### 2.2 问题定义

#### 2.2.1 基于关联规则的标签间依赖

标签关联信息的挖掘是多标签分类问题的研究重点, 现已有多种挖掘标签关联信息的方法应用到多标签学习中<sup>[3]</sup>, 而余弦相似度等<sup>[25]</sup>标签关系的度量方法多为计算标签间相关性, 无法确定标签的相互依赖关系, 为构造标签间依赖关系, 需要一种不对称的标签间关联度量方式.

频繁模式挖掘搜索给定数据集中反复出现的联系, 一个典型的例子是购物篮分析. 该过程通过发现顾客放入“购物篮”中的商品之间的关联, 分析顾客的购物习惯. 如果我们想象全域是商店中商品的集合, 则每种商品有一个布尔变量, 表示该商品是否出现, 每个购物篮可用一个布尔向量表示. 可以分析布尔向量, 得到反映商品频繁关联或同时出现时的购买模式. 这些模式可以用关联规则的形式表示<sup>[26]</sup>.

同样, 我们也可以将多标签分类问题中全体标签看作全域, 每个样本的标签集合可用一个布尔向量表示, 进而

通过分析布尔向量, 得到反映标签频繁关联或同时出现时的模式. 如图 2 为频繁模式思想从购物篮案例到多标签的迁移示意图.

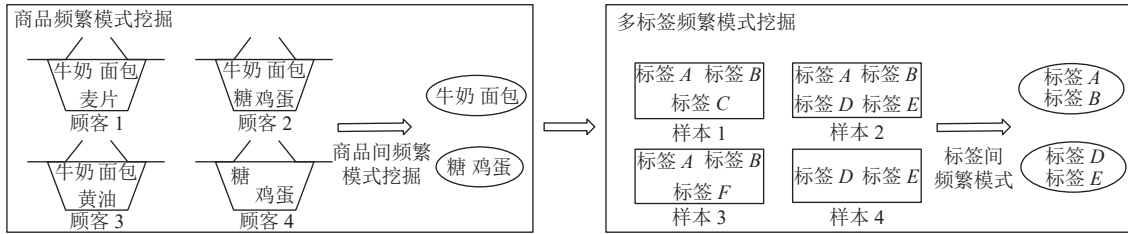


图 2 频繁模式迁移到标签间关联示意图

设  $\gamma = \{l_1, l_2, \dots, l_q\}$  是标签的集合,  $D_Y$  是数据集标签向量的集合, 其中每个标签向量  $Y$  是一个非空标签项集 (项集, 即若干个项的集合), 使得  $Y \subseteq \gamma$ . 设  $A$  是一个标签项集, 标签向量  $Y$  包含  $A$ , 当且仅当  $A \subseteq Y$ . 关联规则是形如  $A \Rightarrow B$  的蕴涵式, 其中  $A \subseteq \gamma, B \subseteq \gamma, A \neq \emptyset, B \neq \emptyset$ , 并且  $A \cap B \neq \emptyset$ . 规则  $A \Rightarrow B$  在数据集  $D_Y$  中成立, 具有支持度  $support$ , 其中  $support$  是数据集  $D_Y$  中标签向量包含  $A \cup B$  (标签项集  $A$  和  $B$  的并集) 的百分比, 即概率  $P(A \cup B)$ . 规则  $A \Rightarrow B$  在数据集  $D_Y$  中具有置信度  $confidence$ , 其中  $confidence$  是数据集  $D_Y$  中包含  $A$  的标签向量同时也包含  $B$  的标签向量的百分比, 即条件概率  $P(B|A)$ .

$$support(A \Rightarrow B) = P(A \cup B) \tag{1}$$

$$confidence(A \Rightarrow B) = P(B|A) \tag{2}$$

最小支持度阈值 ( $min\_sup$ ) 和最小置信度阈值 ( $min\_conf$ ) 是衡量关联规则有效性的最低重要性和最低可靠性, 其中同时满足  $min\_sup$  和  $min\_conf$  的规则为强关联规则<sup>[26]</sup>. 此外我们把标签的集合称为标签项集. 包含  $k$  个标签的标签称为标签  $k$  项集. 标签项集的出现频度是数据集  $D_Y$  中包含标签项集的标签向量数, 简称为标签项集的频度、支持度计数或计数.

由公式 (2), 有:

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} = \frac{support\_count(A \cup B)}{support\_count(A)} \tag{3}$$

其中,  $support\_count(A \cup B)$  是包含标签项集  $A \cup B$  的标签向量数, 而  $support\_count(A)$  是包含标签项集  $A$  的标签向量数.

然而由支持度和置信度推导出的关联规则在特殊情况时具有一定的欺骗性, 有时会产生无趣的关联<sup>[26]</sup>, 因此, 我们引入关联规则中的提升度 ( $lift$ ) 作为对强关联规则的鉴别. 提升度 ( $lift$ ) 是一种简单的相关性度量, 如果  $A$  和  $B$  出现概率独立, 即  $P(A \cup B) = P(A)P(B)$ , 则  $A$  独立于  $B$  出现; 否则,  $A$  和  $B$  是依赖和相关的<sup>[26]</sup>.  $A$  和  $B$  出现之间的提升度可以通过公式 (4) 得到:

$$lift(A, B) = \frac{P(A \cup B)}{P(A)P(B)} = \frac{P(B|A)}{P(B)} = \frac{confidence(A \Rightarrow B)}{support(B)} \tag{4}$$

通过标签一项集、标签二项集出现频度计算标签间提升度  $lift$ , 提升度大于 1, 则标签  $l_i$  和标签  $l_j$  是正相关的, 意味着每一个标签  $l_i$  ( $l_j$ ) 的出现都可能蕴含着标签  $l_j$  ( $l_i$ ) 的出现; 提升度小于 1, 则标签  $l_i$  和标签  $l_j$  是负相关的, 意味着每一个标签  $l_i$  ( $l_j$ ) 的出现可能导致标签  $l_j$  ( $l_i$ ) 不出现.

### 2.2.2 有向无环图和拓扑序列

有向无环图 (directed acyclic graph, DAG) 通常是描述一项工程或系统的进行过程的有效工具. 除最简单的情况外, 几乎所有的工程都可划分为若干个活动的子工程, 而这些子工程之间, 通常受着一定条件的约束, 如其中某些子工程的开始必须在另一些子工程完成之后, 因此 DAG 图常被用来表示事件之间的驱动依赖关系, 管理任务之间的调度. 与子工程之间的约束依赖关系相似, 多标签问题中标签间依赖关系也可看作标签间约束条件. 例如标签  $A$  依赖于标签  $B$ 、标签  $C$ , 即意味着标签  $A$  预测的开始需要在标签  $B$  和标签  $C$  的预测完成后. 利用标签间依赖

关系即可绘制标签间依赖的 DAG 图。

对一个 DAG 图进行拓扑排序, 是将图中所有节点排成一个线性序列, 使得图中任意一对节点  $u$  和  $v$ , 若存在  $u$  指向  $v$  的有向边, 则  $u$  在序列中出现在  $v$  之前。通常, 这样的线性序列称为满足拓扑次序的序列, 简称拓扑序列。拓扑序列常用来确定一个依赖关系集中事件发生的顺序, 通过对 DAG 图进行拓扑排序, 得到拓扑序列, 标签间的依赖关系即可隐含在序列中。

然而在由最小置信度阈值 ( $min\_conf$ ) 更新标签间依赖关系后, 可能存在部分“孤独”标签, 这部分标签不依赖任何标签, 同时其他标签也都不依赖于这部分“孤独”标签 (如图 3 所示, 标签 4、标签 5 即为“孤独”标签)。若以此排序, “孤独”标签可在拓扑序列中任意位置存在, 将会导致拓扑序列数量大大增加。同时为了保证“孤独”标签与其他标签的预测不相互影响, 故将这部分“孤独”标签统一排列在序列前部。

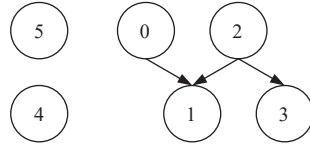


图 3 含有“孤独”标签的标签间依赖关系图

## 2.3 模型建立

### 2.3.1 计算标签间依赖

首先统计数据集标签向量的集合  $D_Y = \{Y_i | 1 \leq i \leq n\}$  中所有标签 ( $L = \{l_1, l_2, \dots, l_q\}$ )、标签间两两组合出现的频度, 得到标签一项集频度  $s_1$  和标签二项集频度  $s_2$ 。定义为:

$$s_1^i = support\_count(l_i) = \sum_{j=1}^n \begin{cases} 1, & Y_j^i = 1 \\ 0, & \text{other} \end{cases} \quad (5)$$

$$s_2^{l_i, l_j} = support\_count(l_i \cup l_j) = \sum_{k=1}^n \begin{cases} 1, & Y_k^i = 1 \text{ and } Y_k^j = 1 \\ 0, & \text{other} \end{cases} \quad (6)$$

接着由公式 (3) 计算得到标签一项集间关联的置信度, 由公式 (4) 计算得到标签一项集间关联的提升度, 定义为:

$$confidence(l_i \Rightarrow l_j) = \frac{support\_count(l_i \cup l_j)}{support\_count(l_i)} = \frac{s_2^{l_i, l_j}}{s_1^i} \quad (7)$$

$$lift(l_i, l_j) = \frac{confidence(l_i \Rightarrow l_j)}{support(l_j)} = \frac{support\_count(l_i \cup l_j) / support\_count(l_i)}{support(l_j)} = \frac{s_2^{l_i, l_j} n}{s_1^i s_1^j} \quad (8)$$

其中,  $n$  为样本数, 提升度大于 1 的标志着标签  $l_i$  和标签  $l_j$  是正相关的, 且二者的正相关度与提升度  $lift$  值成正比, 此时认为二者的依赖关系是显著有效的。设置置信度阈值  $min\_conf$ , 小于置信度阈值的关联规则为弱关联规则, 认为两标签一项集间关联确信度弱, 将其置信度值置为 0。

最后将标签一项集间关联规则及其置信度作为标签间依赖情况及依赖度, 得到不对称的标签间关联度量矩阵, 即为标签依赖矩阵  $W \in \mathbb{R}^{q \times q}$ 。

### 2.3.2 获取拓扑序列

基于标签依赖矩阵即可构造标签间依赖有向无环图, 为确保构造的有向图无环, 将互有依赖的标签 (相互依赖值均大于 0) 的两个依赖度值进行比较, 依赖度弱的值置为 0, 以此消除标签间相互依赖产生的环。但 3 个或 3 个以上的标签间相互依赖仍可能导致的环出现。因此需进行环检测, 并调整  $min\_conf$  值消除图中环结构。有向图环检测流程如下。

- 1) 遍历当前所有节点, 计算每个节点入度 (有向图中某点作为图中边的终点的次数之和)。

2) 统计入度为 0 的节点数量, 若没有, 则图有环结构, 检测结束.

3) 删除入度为 0 的节点及这些节点指向其他节点的关联, 若删除后剩余节点数为 0, 则图无环结构, 检测结束; 否则跳转步骤 1).

最后, 根据标签间依赖有向无环图即可获取标签依赖的拓扑序列. 因 DAG 图特点, 得到的拓扑序列往往不止一条, 故本文设置最大拓扑序列数阈值  $max\_tplen$ , 若拓扑序列数量大于最大拓扑序列数阈值  $max\_tplen$ , 则随机保留其中  $max\_tplen$  条序列. 最终保留  $k$  条拓扑序列, 得到拓扑序列集合  $T$ , 定义如下:

$T = \{t_1, t_2, \dots, t_k\}$ ,  $k \leq max\_tplen$ , 其中,  $t_i$  是  $L = \{l_1, l_2, \dots, l_q\}$  标签集的序列组合.

### 2.3.3 TSECC 模型的训练和预测

TSECC 方法根据标签间依赖情况, 将拓扑序列分为前后两段, 前一段为“孤独”标签项, 后一段为“非孤独”标签项. 前一段使用独立的若干个二元分类器对“孤独”标签项进行训练和预测, 后一段使用分类器链模型进行串联, 充分利用标签间依赖关系, 同时避免了无标签依赖或标签依赖程度较低的“孤独”标签与“非孤独”标签间的相互影响. 图 4 给出了本文方法和传统分类器链方法的对比示意图. 第 4 节给出的实验结果说明, 这样的二元关联和分类器链结合的方法较完全的分类器链方法性能更好.

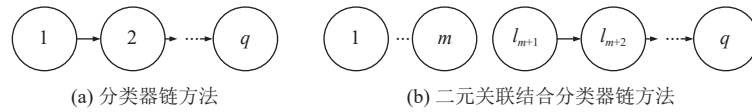


图 4 本文方法与分类器链方法结构示意图

具体而言, 本文方法将得到的拓扑序列  $\{l_{p_i(1)}, l_{p_i(2)}, \dots, l_{p_i(q)}\}$  ( $l_{p_i(j)}$  为第  $i$  条序列, 第  $j$  个位置的标签) 按照标签依赖程度划分为前后两部分  $\{l_{p_i(1)}, \dots, l_{p_i(m)}\}$ 、 $\{l_{p_i(m+1)}, l_{p_i(m+2)}, \dots, l_{p_i(q)}\}$ , 前面部分为无标签依赖或标签依赖程度较低的“孤独”标签, 后面部分为非“孤独”标签. 方法首先对“孤独”标签  $\{l_{p_i(1)}, \dots, l_{p_i(m)}\}$  构建独立的二元分类器, 每一条序列由“孤独”标签得到  $m$  个独立的二元分类器. 记  $BC$  为二元分类算法,  $\beta$  为训练好的二元分类器,  $X$  为训练数据集的特征空间,  $y_{l_{p_i(j)}}$  为标签  $l_{p_i(j)}$  对应的训练数据集的标签向量, 即:

$$\beta_i^j \leftarrow BC(X, y_{l_{p_i(j)}}), j = 1, 2, \dots, m, i = 1, 2, \dots, k \quad (9)$$

接着对非“孤独”标签  $\{l_{p_i(m+1)}, l_{p_i(m+2)}, \dots, l_{p_i(q)}\}$  构建二元分类器, 并将每个分类器的学习结果加入下一个分类器学习的特征空间中, 形成链式结构, 每一条序列得到  $q-m$  个依次串联的二元分类器.

$$\beta_i^j \leftarrow BC(X_i^j, y_{l_{p_i(j)}}), j = m+1, m+2, \dots, q, i = 1, 2, \dots, k \quad (10)$$

其中,  $X_i^j$  为非“孤独”标签构造分类器的特征空间, 定义为:

$$X_i^j = X_i^{j-1} \cup y_{l_{p_i(j-1)}} \quad (11)$$

## 2.4 算法实现

### 2.4.1 计算标签间依赖算法

标签间依赖关系是形成拓扑序列的依据. 算法 1 给出了利用关联规则计算标签间依赖关系及依赖度的过程.

**算法 1.** 获取标签依赖矩阵  $W$ .

输入: 训练数据集标签向量集合  $D_Y = \{Y_i | 1 \leq i \leq n\}$ , 置信度阈值  $min\_conf$ ;

输出: 标签依赖矩阵  $W \in \mathbb{R}^{q \times q}$ .

- 1) 初始化标签一项集频度  $s_1 \in O^q$  和标签二项集频度  $s_2 \in O^{q \times q}$  空间,  $O$  代表全 0 向量
- 2) 初始化标签依赖矩阵  $W \in \mathbb{R}^{q \times q}$
- 3) for  $i \leftarrow 1$  to  $q$
- 4) for  $j \leftarrow 1$  to  $n$

---

```

5)   if  $Y_j^i = 1$  then  $s_1^i \leftarrow s_1^i + 1$  /*计算标签一项集频度*/
6)   for  $k \leftarrow 1$  to  $q$ 
7)     if  $Y_j^i = 1$  and  $Y_j^k = 1$  then  $s_1^{i,k} \leftarrow s_1^{i,k} + 1$  /*计算标签二项集频度*/
8)   endfor
9) endfor
10) for  $i \leftarrow 1$  to  $q$ 
11)   for  $j \leftarrow 1$  to  $q$ 
12)     if  $s_2^{i,j} / s_1^i > \min\_conf$  and  $s_2^{i,j} n / s_1^i s_1^j > 1$  then  $w_{ij} \leftarrow s_2^{i,j} / s_1^i$ 
13)   endfor
14) endfor

```

---

#### 2.4.2 获取拓扑全序列算法

对标签依赖矩阵  $W$  进行拓扑排序, 其拓扑序列蕴含了标签间的依赖情况. 算法 2 给出了由标签依赖矩阵  $W$  生成拓扑排序全序列的过程.

---

#### 算法 2. 获取拓扑排序全序列算法.

---

输入: 标签依赖矩阵  $W \in \mathbb{R}^{q \times q}$ , 最大拓扑序列数阈值  $max\_tplen$ ;

输出: 拓扑排序全序列  $T = \{t_1, t_2, \dots, t_k\}$ .

---

```

1) 初始化空序列数组  $T$  存储所有拓扑序列
2) 初始化空数组  $preList$  存储当前序列
3)  $topologicalSort(W, preList, T)$ 
4)   if  $length(preList) = q$  then  $T \leftarrow T \cup preList$ 
5)   if  $length(T) = max\_tplen$  then end
6)   for  $i \leftarrow 1$  to  $q$ 
7)     if  $sum(w_i) = 0$  and  $i$  not in  $preList$  then  $preList \leftarrow preList \cup i$ 
8)     for  $j \leftarrow 1$  to  $q$ 
9)        $w_{ji} \leftarrow 0$ 
10)    endfor
11)     $topologicalSort(W, preList, T)$ 
12) endfor

```

---

#### 2.4.3 改进的分类器链模型的训练和预测算法

考虑到“孤独”标签易对标签关联信息传递造成影响, 本文 TSECC 方法将“孤独”标签与“非孤独”标签分离, 使用若干个独立的二元分类器训练. 算法 3 和算法 4 分别给出了本文方法的训练和预测过程.

---

#### 算法 3. TSECC 训练过程.

---

输入: 拓扑排序全序列  $T = \{t_1, t_2, \dots, t_k\}$ , 训练数据集  $D = \{(X_i, Y_i) | 1 \leq i \leq n\}$ , “孤独”标签数量  $m$ ;

输出:  $k \times q$  个分类器  $\beta_i^j, i = 1, 2, \dots, k, j = 1, 2, \dots, q$ .

---

```

1) for  $i \leftarrow 1$  to  $k$ 
2)   for  $j \leftarrow 1$  to  $m$ 
3)      $\beta_i^j \leftarrow BC(X, y_{t_{p_i(j)}})$ 

```

---

- 
- 
- 4) endfor
  - 5) for  $j \leftarrow m+1$  to  $q$
  - 6)  $X_i^j \leftarrow X_i^{j-1} \cup y_{l_{p_i(j-1)}}$
  - 7)  $\beta_i^j \leftarrow BC(X_i^j, y_{l_{p_i(j)}})$
  - 8) endfor
  - 9) endfor
- 
- 

**算法 4.** TSECC 预测过程.

输入: 待预测数据集  $D' = \{(X_i^i) | 1 \leq i \leq n'\}$ ,  $k \times q$  个分类器  $\beta_i^j$ , 拓扑排序全序列  $T = \{t_1, t_2, \dots, t_k\}$ , “孤独”标签数量  $m$ ;  
 输出: 预测结果  $y' = \{(Y_i^i) | 1 \leq i \leq n'\}$ .

- 
- 1) 初始化  $k$  个  $y'$  空间  $Z$
  - 2) for  $i \leftarrow 1$  to  $k$
  - 3) for  $j \leftarrow 1$  to  $m$
  - 4)  $Z_i^j \leftarrow \beta_i^j(X_i^i)$
  - 5) endfor
  - 6) for  $j \leftarrow m+1$  to  $q$
  - 7)  $(X_i^i)^j \leftarrow (X_i^i)^{j-1} \cup Z_i^j$
  - 8)  $Z_i^j \leftarrow \beta_i^j((X_i^i)^j, y_{l_{p_i(j)}})$
  - 9) endfor
  - 10) endfor
  - 11)  $y' \leftarrow \lceil [\bar{Z}_i^1 - 1/2], [\bar{Z}_i^2 - 1/2], \dots, [\bar{Z}_i^q - 1/2] \rceil$
- 

### 3 实验与结果

#### 3.1 实验数据

本文对 5 个多标签数据集进行了实验, 以探究本文算法的性能. 这些数据集来自多标签的不同应用领域: Flags, Scene 来自图片分类, Birds, Emotions 来自音乐标注, Yeast 来自生物基因功能预测. 这些数据集可以在开源项目 mulan<sup>[27]</sup>的主页 (<http://mlkd.csd.auth.gr/multilabel.html>) 下载获得, 表 1 给出了数据集的详细统计信息.

表 1 实验数据集

数据集	领域	示例数目	特征维度	标签数目	标签基数	标签密度	标签多样性
Flags	图片	194	19	7	3.392	0.485	54
Scene	图片	2 407	294	6	1.074	0.179	15
Birds	音乐	645	260	19	1.014	0.053	133
Emotions	音乐	593	72	6	1.869	0.311	27
Yeast	生物	2 417	103	14	4.237	0.303	198
CAL500	音乐	502	68	174	26.044	0.150	502

标签基数 (label cardinality), 每个样本相关标签的平均个数:

$$LCard(D) = \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^q y_i^j \quad (12)$$



标签密度 (label density), 标签基数相对于标签数目的比例:

$$LDen(D) = \frac{1}{q} LCard(D) \quad (13)$$

标签多样性 (label distinct), 样本相关标签集合的总数:

$$LDis(D) = |\{y | \exists x : (x, y) \in D\}| \quad (14)$$

### 3.2 评价指标及对比模型

本文采用多标签学习领域 3 个常用的指标 hammingloss, macro-F1, micro-F1<sup>[1]</sup>来衡量方法的预测性能. 定义如下.

1) hammingloss: 衡量错分的标签比例, 正确标签没有被预测以及错误标签被预测的标签占比.

2) macro-F1: 每个标签 F1 指标的平均.

3) micro-F1: 计算所有样本在所有标记上的总体精准度 (micro-P) 和召回率 (micro-R), 再计算 F1 分数.

用于本文对比的方法为 4 个分类器链相关算法和 2 个其他多标签算法.

1) BR (binary relevant) 方法, 该方法不考虑标签间关系, 独立训练每个标签的二类分类器<sup>[3]</sup>.

2) CC (classifier chains) 方法, 分类器链算法<sup>[4,5]</sup>.

3) ECC (ensemble classifier chains) 方法, 集成分类器链算法<sup>[4,5]</sup>.

4) CCE (classifier circle) 方法, 基于分类器链算法改进的分类器圈算法, 使用圈结构避免标签序列的不确定<sup>[18]</sup>.

5) LSF-CC (label specific features-based classifier chains) 方法, 基于标签特征选择的分类器链算法, 通过特征选择降低冗余特征对标签传递的干扰<sup>[22]</sup>.

6) ML-KNN (multi-kabel K-nearest neighbor) 方法, 该方法拓展 K 近邻方法用于处理多标签学习问题<sup>[12]</sup>.

7) TSECC (topological sequence-ensemble classifier chains) 方法, 即本文的基于关联规则和拓扑序列的分类器链算法.

### 3.3 实验结果

实验过程中, 对于每个数据集, 本文随机选取 70% 的实例作为训练样本, 其余 30% 的实例作为测试样本. 为降低随机性影响, 本文针对每个数据集的每个方法均重复实验 30 次, 计算各项指标平均值及标准差作为实验最终结果.

本文所有实验在 Python 平台完成, 分类器链及其改进算法使用的二类分类器采用 AdaBoost、SVM 等基分类器, 基分类器及评价指标 (hammingloss、macro-F1、micro-F1) 均由 Python 平台下的 scikit-learn 工具包实现.

表 2 给出本文算法与 BR、CC、ECC、CCE、LSF-CC 和 ML-KNN 这 6 种对比算法 (分类器链及其改进算法基于 AdaBoost 分类器) 在 5 个数据集上关于 hammingloss、macro-F1 和 micro-F1 这 3 个评价指标的平均结果和标准差. 评价指标中, hammingloss 指标为错误预测比例, 该值越小算法性能越好; macro-F1 和 micro-F1 指标为算法“查全率”和“查准率”的综合指标, 该值越大算法性能越好. 其中, 加粗字体表示该算法在该数据集及对应指标下取得最优值 (成对 *t* 检验根据 95% 置信度), ↑ (↓) 表示该评价指标越大 (越小) 对应方法性能越好.

分析表 2 中数据可知, TSECC 在各项数据集中均取得了相对较好的性能, 其中在 Emotions、Flags、Birds 数据集上的性能尤为突出, 原因是这 3 个数据集皆具有标签关联程度高、标签维度低的特点, TSECC 算法充分利用了隐含在标签序列中的依赖关系. 在 Scene 数据集中, BR 算法取得了最好的性能, 分类器链及相关改进算法性能大致相同, 原因是 Scene 数据集标签关联程度相对较弱. 而在 Yeast 数据集中, 利用样本相似性进行标签传播的 ML-KNN 综合性能最优, 同样未考虑标签相关的 BR 算法虽在 hammingloss 指标较好, 但在综合考虑查全率和查准率的 macro-F1 和 micro-F1 指标中性能较差, 原因是 Yeast 数据集标签维度较高, 虽有较强的标签关联程度, 但标签次序的排列组合数量大, 依赖标签学习序列的分类器链相关算法随机性强, 此时不考虑标签学习序列的 ML-KNN 和 CCE 则拥有更好的性能, 而 LSF-CC 的特征选择策略在标签维度较大时, 能有效增强标签关联信息的传递, 因此也取得了良好的性能.

表 2 TSECC 与对比方法实验结果 (AdaBoost 基分类器)

评价指标	数据集	Emotions	Flags	Scene	Yeast	Birds
hammingloss↓	TSECC	<b>0.2013±0.0104(1)</b>	<b>0.2796±0.0064(1)</b>	0.0943±0.0017(3)	0.2063±0.0032(5)	<b>0.0395±0.0003(1)</b>
	ECC	0.2182±0.0116(5)	0.2873±0.0126(2)	0.0989±0.0037(6)	0.2106±0.0026(6)	0.0400±0.0004(5)
	CC	0.2282±0.0111(6)	0.2883±0.0124(4)	0.0976±0.0040(5)	0.2142±0.0041(7)	0.0396±0.0005(3)
	BR	0.2141±0.0099(3)	0.2962±0.0054(6)	<b>0.0894±0.0014(1)</b>	<b>0.1979±0.0019(1)</b>	0.0403±0.0001(6)
	ML-KNN	0.2538±0.0112(7)	0.3174±0.0132(7)	0.0989±0.0058(7)	0.1984±0.0046(2)	0.1145±0.0003(7)
	CCE	0.2182±0.0066(4)	0.2896±0.0090(5)	0.0936±0.0016(2)	0.1995±0.0047(3)	0.0395±0.0008(2)
	LSF-CC	0.2057±0.0121(2)	0.2879±0.0098(3)	0.0969±0.0034(4)	0.2023±0.0041(4)	0.0399±0.0006(4)
macro-F1↑	TSECC	<b>0.6564±0.0191(1)</b>	0.6341±0.0073(2)	0.7372±0.0041(2)	0.3578±0.0064(4)	<b>0.3329±0.0088(1)</b>
	ECC	0.6360±0.0229(3)	<b>0.6343±0.0168(1)</b>	0.7284±0.0099(6)	0.3532±0.0083(5)	0.3222±0.0069(5)
	CC	0.6087±0.0222(6)	0.6160±0.0163(5)	0.7358±0.0105(5)	0.3436±0.0079(6)	0.3266±0.0089(3)
	BR	0.6354±0.0219(4)	0.6044±0.0056(6)	<b>0.7485±0.0072(1)</b>	0.3290±0.0071(7)	0.3166±0.0021(6)
	ML-KNN	0.4018±0.0203(7)	0.4572±0.0153(7)	0.6916±0.0059(7)	<b>0.3743±0.0069(1)</b>	0.0111±0.0114(7)
	CCE	0.6352±0.0171(5)	0.6265±0.0111(3)	0.7422±0.0047(3)	0.3592±0.0073(3)	0.3284±0.0016(2)
	LSF-CC	0.6531±0.0216(2)	0.6263±0.0101(4)	0.7361±0.0062(4)	0.3648±0.0078(2)	0.3248±0.0084(4)
micro-F1↑	TSECC	<b>0.6635±0.0208(1)</b>	<b>0.7154±0.0063(1)</b>	0.7275±0.0034(2)	0.6335±0.0058(6)	<b>0.5067±0.0041(1)</b>
	ECC	0.6435±0.0214(5)	0.7058±0.0131(2)	0.7233±0.0095(6)	0.6398±0.0074(3)	0.4949±0.0054(5)
	CC	0.6228±0.0208(6)	0.7029±0.0125(4)	0.7258±0.0103(5)	0.6171±0.0078(7)	0.5032±0.0068(2)
	BR	0.6454±0.0214(4)	0.6956±0.0048(6)	<b>0.7407±0.0027(1)</b>	0.6342±0.0032(5)	0.4996±0.0015(4)
	ML-KNN	0.4761±0.0207(7)	0.6648±0.0142(7)	0.6986±0.0064(7)	0.6375±0.0070(4)	0.0738±0.0709(7)
	CCE	0.6574±0.0121(3)	0.7037±0.0084(3)	0.7259±0.0042(4)	0.6417±0.0092(2)	0.5006±0.0026(3)
	LSF-CC	0.6628±0.0213(2)	0.7021±0.0095(5)	0.7267±0.0071(3)	<b>0.6454±0.0082(1)</b>	0.4941±0.0063(6)

为了更直观地展示本文算法与各对比算法的实验效果, 本文对比分析了每个算法在 3 项评价指标和 5 个数据集上的平均排名. 图 5 展示了每个算法在 3 项评价指标上的平均排名, 例如, TSECC 算法在 5 个数据集的 hammingloss 指标排名分别为 {1, 1, 3, 5, 1}, 那么 TSECC 算法在 hammingloss 的平均排名为  $(1+1+3+5+1)/5=2.2$ . 图 6 展示了每个算法在 5 个数据集上的平均排名, 例如, TSECC 算法在 Scene 数据集上关于 hammingloss, macro-F1, micro-F1 这三项指标的排名分别为 {3, 2, 2}, 则在 Scene 数据集上的平均排名为  $(3+2+2)/3=2.34$ .

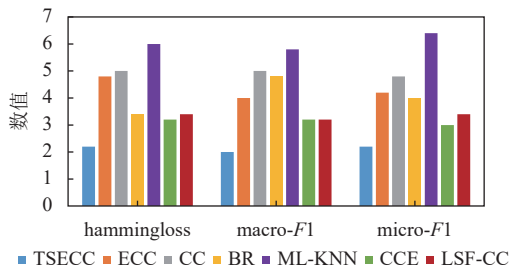


图 5 各算法在 3 个指标的平均排名 (柱形图越低, 性能越好)

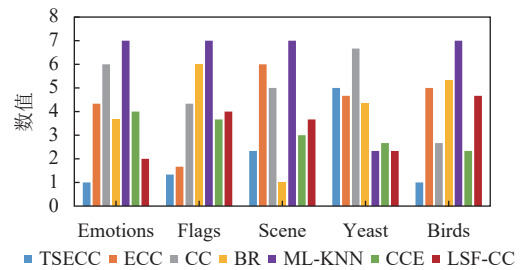


图 6 各算法在 5 个数据集上的平均排名 (柱形图越低, 性能越好)

综合来看, 在 5 个数据集的验证中, TSECC 算法在 3 个数据集的 hammingloss 和 micro-F1 指标上取得了最好的性能, 在两个数据集的 macro-F1 指标上取得了最好的性能, 且在 3 项指标上均取得最小的平均排名, 这一结果验证了本文方法的有效性.

### 3.3.1 基分类器影响实验

为了比较分类器链及其改进算法在采用不同基分类器下的性能, 表 3 给出了分类器链及其改进算法在以 SVM (liner 核函数) 为基分类器下的实验结果, 与上述实验设置相同, 随机选取 70% 的实例作为训练样本, 其余 30% 的实例作为测试样本, 各组实验均重复进行 30 次, 计算各算法在 5 个数据集下的 3 个指标的平均值及标准

差. 实验结果显示, 对比表 2 (以 AdaBoost 为基分类器) 实验结果, TSECC 算法由于更有效地学习提取了标签关联信息, 在各个数据集表现稳定. 在标签关联程度高的 Emotions、Flags 和 Birds 数据集上的综合性能皆优于其他对比算法, 而在标签关联程度不高的 Scene 数据集上, 不考虑标签学习次序的分类器圈算法 (CCE) 和考虑标签相关的二元关联算法 (BR) 取得了更好的性能. 在标签维度较大的 Yeast 数据集上, 受阶乘级的标签序列数量影响, 分类器链及其改进算法性能大体相同, TSECC 算法略优于其他方法. 该组实验中各算法的平均排名如图 7、图 8 所示.

表 3 TSECC 与对比方法实验结果 (SVM 基分类器)

评价指标	数据集	Emotions	Flags	Scene	Yeast	Birds
hammingloss↓	TSECC	<b>0.2102±0.0064(1)</b>	<b>0.2832±0.0082(1)</b>	0.1146±0.0032(3)	0.2010±0.0052(3)	<b>0.0628±0.0008(1)</b>
	ECC	0.2282±0.0040(6)	0.2926±0.0119(2)	0.1183±0.0049(5)	0.2079±0.0025(5)	0.0759±0.0006(4)
	CC	0.2245±0.0069(5)	0.2996±0.0136(5)	0.1201±0.0072(6)	0.2118±0.0049(6)	0.0792±0.0008(5)
	BR	0.2166±0.0093(4)	0.3039±0.0091(6)	<b>0.1035±0.0051(1)</b>	<b>0.1979±0.0013(1)</b>	0.0834±0.0003(6)
	CCE	0.2143±0.0076(3)	0.2951±0.0117(3)	0.1074±0.0043(2)	0.1984±0.0017(2)	0.0691±0.0011(2)
	LSF-CC	0.2129±0.0097(2)	0.2962±0.0114(4)	0.1175±0.0038(4)	0.2057±0.0063(4)	0.0718±0.0009(3)
macro-F1↑	TSECC	<b>0.6391±0.0142(1)</b>	<b>0.6176±0.0087(1)</b>	0.6809±0.0068(2)	<b>0.3545±0.0053(1)</b>	<b>0.3193±0.0097(1)</b>
	ECC	0.6121±0.0171(5)	0.6004±0.0142(4)	0.6741±0.0095(4)	0.3515±0.0100(3)	0.3022±0.0082(4)
	CC	0.6246±0.0199(3)	0.5942±0.0176(5)	0.6692±0.0083(6)	0.3524±0.0226(2)	0.2894±0.0105(6)
	BR	0.5917±0.0182(6)	0.5914±0.0106(6)	0.6697±0.0037(5)	0.3290±0.0032(6)	0.2961±0.0076(5)
	CCE	0.6244±0.0172(4)	0.6053±0.0163(2)	<b>0.6962±0.0062(1)</b>	0.3373±0.0067(5)	0.3128±0.0063(2)
	LSF-CC	0.6301±0.0195(2)	0.6764±0.0082(3)	0.6764±0.0082(3)	0.3471±0.0087(4)	0.3082±0.0069(3)
micro-F1↑	TSECC	<b>0.6495±0.0174(1)</b>	<b>0.6954±0.0092(1)</b>	0.6727±0.0062(3)	0.6445±0.0083(3)	<b>0.4983±0.0058(1)</b>
	ECC	0.6326±0.0111(5)	0.6841±0.0153(2)	0.6688±0.0093(5)	0.6470±0.0036(2)	0.4728±0.0067(6)
	CC	0.6398±0.0137(4)	0.6748±0.0169(5)	0.6646±0.0104(6)	0.6413±0.0095(4)	0.4891±0.0081(2)
	BR	0.6236±0.0165(6)	0.6617±0.0063(6)	0.6790±0.0079(2)	0.6342±0.0027(6)	0.4767±0.0068(5)
	CCE	0.6418±0.0147(3)	0.6813±0.0083(3)	<b>0.6910±0.0086(1)</b>	0.6402±0.0046(5)	0.4827±0.0057(4)
	LSF-CC	0.6424±0.0192(2)	0.6786±0.0130(4)	0.6693±0.0086(4)	<b>0.6486±0.0078(1)</b>	0.4863±0.0079(3)

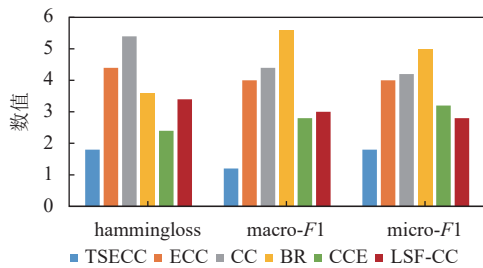


图 7 分类器链及改进算法 (SVM 为基分类器) 在 3 个指标上的平均排名

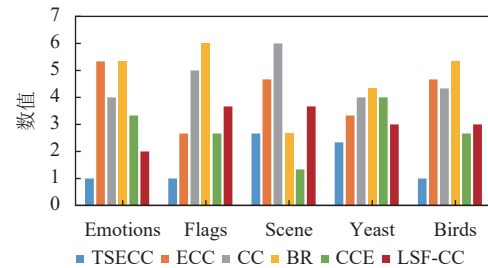


图 8 分类器链及改进算法 (SVM 为基分类器) 在 5 个数据集上的平均排名

上述实验结果表明, 本文 TSECC 算法在多个数据集中具有良好的性能, 且整体性能优于其他对比算法. TSECC 算法在 Emotions、Flags 和 Birds 数据集上均获得了最佳性能, 在 Scene 和 Yeast 数据集上也取得了相对其他对比算法更好的性能, 这些实验结果证明了本文算法在不同基分类器下均表现良好, 模型鲁棒性强.

### 3.3.2 算法稳定性实验

数据集的标签维度过高时, 分类器链及其改进算法面对阶乘级增长的标签序列, 容易出现性能不稳定的问题, 导致预测随机性强. TSECC 算法不同于分类器链相关算法随机的选取学习次序, 而是通过学习标签间依赖信息计算蕴含标签关联信息的标签学习次序, 因此拥有更稳定的性能. 本文对比了 TSECC 和 CC、ECC 方法在 CAL500 数据集 (标签维度 174) 上的表现, 各组实验均重复进行 30 次, 计算各算法在 CAL500 数据集下 3 个指标的平均值

及标准差. 实验结果表明, 面对标签维度较高的数据集时, 高维标签削弱了链式传播关联信息的能力, 分类器链及其改进算法预测准确度及性能大体相同, 但 TSECC 方法的主动计算选取学习次序而非随机生成次序的策略使其拥有最优的稳定性, 表 4 给出了 3 个算法在 CAL500 数据集上的实验结果, 为了更直观的展示实验效果, 图 9 给出了 3 个算法在该数据集上的稳定性 (标准差) 对比图. 实验结果表明, TSECC 算法对比 CC、ECC 算法具有更好的稳定性.

表 4 CAL500 数据集实验结果

评价指标	TSECC	ECC	CC
hammingloss↓	0.13521±0.00009	0.13571±0.00022	0.13502±0.00079
macro-F1↑	0.05632±0.00225	0.05551±0.00494	0.05691±0.00720
micro-F1↑	0.32917±0.00831	0.33221±0.01698	0.32848±0.01102

### 3.3.3 “孤独”标签效果实验

针对标签依赖中可能出现的“孤独”标签问题, TSECC 算法采用了二元关联结合分类器链的方法, 将可能出现的“孤独”标签排在次序前列, 利用标签数量个独立的二元分类器进行训练和预测, 剩下的标签则采用分类器链模型训练和预测. 为验证“孤独”标签策略的有效性, 本文对比了 TSECC 方法和 TSECC-NG (未使用“孤独”标签策略), 图 10-图 12 给出了 TSECC 方法和 TSECC-NG 方法在 3 个指标上的对比实验结果. 实验结果表明, 采用二元关联结合分类器链的 TSECC 方法总是略优于未使用“孤独”标签策略的 TSECC-NG 方法, 原因是“孤独”标签的分离保证了无标签依赖或标签依赖程度较低的标签不会对其他标签传递的依赖信息造成干扰.

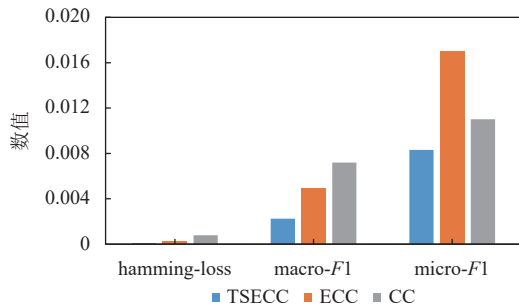


图 9 CAL500 数据集上 3 种方法稳定性 (标准差) 对比 (柱形图越低, 稳定性能越好)

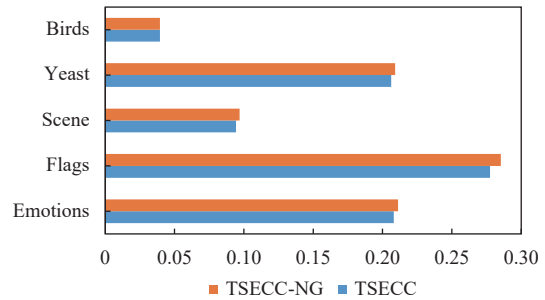


图 10 hammingloss 性能结果

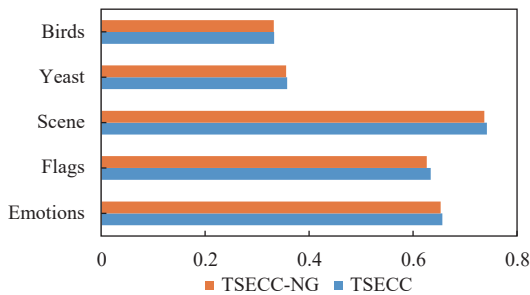


图 11 macro-F1 性能结果

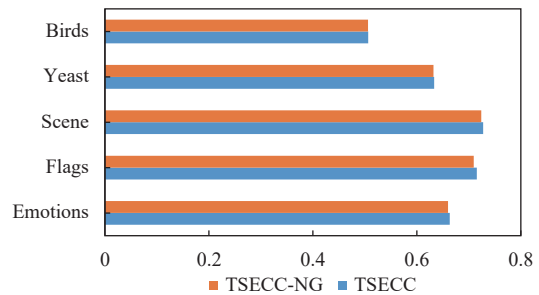


图 12 micro-F1 性能结果

### 3.3.4 模型参数分析

分类器链相关算法通过集成多条不同标签学习次序的分类器链以弱化单条链存在的强随机性. 为进一步验证该集成数量参数对算法性能以及算法稳定性的影响, 本文针对 TSECC 和 ECC 算法中集成分类器链数量参数进行了测试实验. 实验设置与上文实验方法相同, 重复 30 轮, 计算在不同分类器链数量时两个算法在 hammingloss、

macro-F1 及 micro-F1 指标的平均性能, 图 13 给出了实验结果. 由图 13 可知, 在数据集标签维度不高时, ECC 算法随机性较大, TSECC 算法具有更好的稳定性, 参数改变时性能波动较小, 原因是 TSECC 算法能较稳定的通过计算标签间依赖关系来生成标签学习次序; 而在数据集标签维度较高时, TSECC 和 ECC 算法面对阶乘量级的标签序列, 均具有一定的随机性, 但对比标签学习次序完全随机的 ECC 算法, TSECC 算法的性能和稳定性均优于 ECC 算法. 该实验验证了对比 ECC 算法, TSECC 具有更良好的稳定性, 对集成分类器链数量参数不敏感.

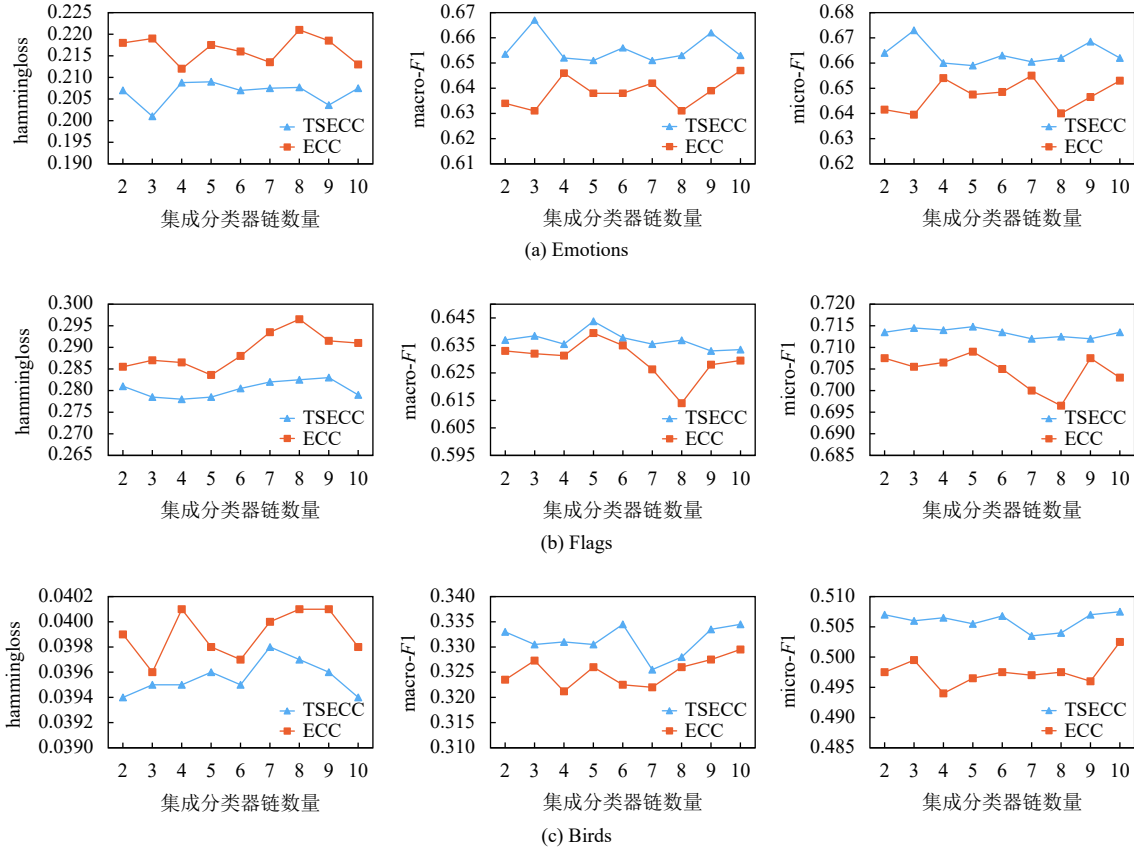


图 13 TSECC 算法与 ECC 算法中分类器链集成数对性能影响实验结果

### 3.3.5 时间复杂度分析

TSECC 算法包括计算标签间依赖度, 生成标签学习次序, 模型训练和预测 4 个部分. 计算标签间依赖度和生成标签学习次序的时间复杂度分别为  $O(q \cdot q \cdot n)$  和  $O(q + e)$ , 其中  $e$  为构建的有向无环图中边的条数. 另外 TSECC 算法的模型训练和预测部分的时间复杂度分别为  $k \cdot \left[ O_{\beta}(n, d) + \sum_{i=1}^{q-m} O_{\beta}(n, d + i - 1) \right]$  和  $k \cdot \left[ O'_{\beta}(d) + \sum_{i=1}^{q-m} O'_{\beta}(d + i - 1) \right]$ , 其中  $O_{\beta}(\cdot, \cdot)$  和  $O'_{\beta}(\cdot)$  是二元分类器训练和预测的时间复杂度; ECC 算法的模型训练和预测部分的时间复杂度分别为  $k \cdot \sum_{i=1}^q O_{\beta}(n, d + i - 1)$  和  $k \cdot \sum_{i=1}^q O'_{\beta}(d + i - 1)$ . 因此, TSECC 算法在模型训练和预测部分对比 ECC 算法均取得了相同或更少的时间复杂度, 这取决于当前数据集中“孤独”标签的数量  $m$ , 当  $m$  不为 0 时, TSECC 取得更小的时间复杂度, 否则两者时间开销相当.

## 4 总结

本文提出一种基于关联规则和拓扑序列的分类器链算法, 解决了分类器链及其相关改进算法在面对阶乘量级

的标签序列时性能不佳且随机性强的问题. 算法把频繁模式中关联规则的思想迁移到标签关联信息的挖掘中, 将基于标签间依赖关系的拓扑序列作为分类器链的学习次序, 降低了模型的随机性, 提高了分类性能. 此外将“孤独”标签独立出分类器链模型单独训练, 提高了模型的容错. 实验结果表明, TSECC 方法取得了比分类器链及其相关改进算法更好的性能.

## References:

- [1] Zhang ML, Zhou ZH. A review on multi-label learning algorithms. *IEEE Trans. on Knowledge and Data Engineering*, 2014, 26(8): 1819–1837. [doi: [10.1109/TKDE.2013.39](https://doi.org/10.1109/TKDE.2013.39)]
- [2] Tsoumakas G, Katakis I. Multi-label classification: An overview. *Int'l Journal of Data Warehousing and Mining*, 2007, 3(3): 1. [doi: [10.4018/jdwm.2007070101](https://doi.org/10.4018/jdwm.2007070101)]
- [3] Boutell MR, Luo JB, Shen XP, Brown CM. Learning multi-label scene classification. *Pattern Recognition*, 2004, 37(9): 1757–1771. [doi: [10.1016/j.patcog.2004.03.009](https://doi.org/10.1016/j.patcog.2004.03.009)]
- [4] Read J, Pfahringer B, Holmes G, Frank E. Classifier chains for multi-label classification. In: *Proc. of the 2009 European Conf. on Machine Learning and Knowledge Discovery in Databases*. Bled: Springer, 2009. 254–269. [doi: [10.1007/978-3-642-04174-7\\_17](https://doi.org/10.1007/978-3-642-04174-7_17)]
- [5] Read J, Pfahringer B, Holmes G, Frank E. Classifier chains for multi-label classification. *Machine Learning*, 2011, 85(3): 333–359. [doi: [10.1007/s10994-011-5256-5](https://doi.org/10.1007/s10994-011-5256-5)]
- [6] Dembczynski K, Cheng WW, Hüllermeier E. Bayes optimal multilabel classification via probabilistic classifier chains. In: *Proc. of the 27th Int'l Conf. on Machine Learning*. Haifa: ACM, 2010. 279–286. [doi: [10.5555/3104322.3104359](https://doi.org/10.5555/3104322.3104359)]
- [7] Wang TS, Liu L, Liu NW, Zhang HX, Zhang L, Feng SS. A multi-label text classification method via dynamic semantic representation model and deep neural network. *Applied Intelligence*, 2020, 50(8): 2339–2351. [doi: [10.1007/s10489-020-01680-w](https://doi.org/10.1007/s10489-020-01680-w)]
- [8] Dai JH, Chen JL, Liu Y, Hu H. Novel multi-label feature selection via label symmetric uncertainty correlation learning and feature redundancy evaluation. *Knowledge-based Systems*, 2020, 207: 106342. [doi: [10.1016/j.knsys.2020.106342](https://doi.org/10.1016/j.knsys.2020.106342)]
- [9] Lin LY, Liu B, Zheng X, Xiao YS, Liu ZJ, Cai H. An efficient multi-label learning method with label projection. *Knowledge-based Systems*, 2020, 207: 106298. [doi: [10.1016/j.knsys.2020.106298](https://doi.org/10.1016/j.knsys.2020.106298)]
- [10] Xie J, Yu L, Zhu L, Duan GL. Conditional entropy based classifier chains for multi-label classification. *Neurocomputing*, 2019, 335: 185–194. [doi: [10.1016/j.neucom.2019.01.039](https://doi.org/10.1016/j.neucom.2019.01.039)]
- [11] Sun L, Kudo M. Multi-label classification by polytree-augmented classifier chains with label-dependent features. *Pattern Analysis and Applications*, 2019, 22(3): 1029–1049. [doi: [10.1007/s10044-018-0711-6](https://doi.org/10.1007/s10044-018-0711-6)]
- [12] Zhang ML, Zhou ZH. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 2007, 40(7): 2038–2048. [doi: [10.1016/j.patcog.2006.12.019](https://doi.org/10.1016/j.patcog.2006.12.019)]
- [13] Elisseeff A, Weston J. A kernel method for multi-labelled classification. In: *Proc. of the 14th Int'l Conf. on Neural Information Processing Systems: Natural and Synthetic*. Vancouver: ACM, 2001. 681–687. [doi: [10.5555/2980539.2980628](https://doi.org/10.5555/2980539.2980628)]
- [14] Ghamrawi N, McCallum A. Collective multi-label classification. In: *Proc. of the 14th ACM Int'l Conf. on Information and Knowledge Management*. Bremen: ACM, 2005. 195–200. [doi: [10.1145/1099554.1099591](https://doi.org/10.1145/1099554.1099591)]
- [15] Zhang YJ, Miao DQ, Zhang ZF, *et al.* A three-way selective ensemble model for multi-label classification. *Int'l Journal of Approximate Reasoning*, 2018, 103: 394–413. [doi: [10.1016/j.ijar.2018.10.009](https://doi.org/10.1016/j.ijar.2018.10.009)]
- [16] Huang KH, Lin HT. Cost-sensitive label embedding for multi-label classification. *Machine Learning*, 2017, 106(9): 1725–1746. [doi: [10.1007/s10994-017-5659-z](https://doi.org/10.1007/s10994-017-5659-z)]
- [17] Madjarov G, Kocev D, Gjorgjevikj D, Džeroski S. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 2012, 45(9): 3084–3104. [doi: [10.1016/j.patcog.2012.03.004](https://doi.org/10.1016/j.patcog.2012.03.004)]
- [18] Wang SB, Li YF. Classifier circle method for multi-label learning. *Ruan Jian Xue Bao/Journal of Software*, 2015, 26(11): 2811–2819 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4908.htm> [doi: [10.13328/j.cnki.jos.004908](https://doi.org/10.13328/j.cnki.jos.004908)]
- [19] Li N, Pan ZS, Zhou XY. Classifier chain algorithm based on multi-label importance rank. *Pattern Recognition and Artificial Intelligence*, 2016, 29(6): 567–575 (in Chinese with English abstract). [doi: [10.16451/j.cnki.issn1003-6059.201606011](https://doi.org/10.16451/j.cnki.issn1003-6059.201606011)]
- [20] Wang ZW, Wang TL, Wan BT, Han MJ. Partial classifier chains with feature selection by exploiting label correlation in multi-label classification. *Entropy*, 2020, 22(10): 1143. [doi: [10.3390/e22101143](https://doi.org/10.3390/e22101143)]
- [21] Hu TL, Wang HB, Yin WD. Multi-label news classification algorithm based on deep bi-directional classifier chains. *Journal of Zhejiang University (Engineering Science)*, 2019, 53(11): 2110–2117 (in Chinese with English abstract). [doi: [10.3785/j.issn.1008-973X.2019.11.008](https://doi.org/10.3785/j.issn.1008-973X.2019.11.008)]

- [22] Weng W, Wang DH, Chen CL, Wen J, Wu SX. Label specific features-based classifier chains for multi-label classification. IEEE Access, 2020, 8: 51265–51275. [doi: [10.1109/ACCESS.2020.2980551](https://doi.org/10.1109/ACCESS.2020.2980551)]
- [23] Liu B, Tsoumakas G. Dealing with class imbalance in classifier chains via random undersampling. Knowledge-based Systems, 2020, 192: 105292. [doi: [10.1016/j.knosys.2019.105292](https://doi.org/10.1016/j.knosys.2019.105292)]
- [24] Sun L, Kudo M. Optimization of classifier chains via conditional likelihood maximization. Pattern Recognition, 2018, 74: 503–517. [doi: [10.1016/j.patcog.2017.09.034](https://doi.org/10.1016/j.patcog.2017.09.034)]
- [25] Wu B Y, Jia F, Liu W, Ghanem B, Lyu SW. Multi-label learning with missing labels using mixed dependency graphs. Int'l Journal of Computer Vision, 2018, 126(8): 875–896. [doi: [10.1007/s11263-018-1085-3](https://doi.org/10.1007/s11263-018-1085-3)]
- [26] Han JW, Kamber M, Pei J. Data Mining: Concepts and Techniques. 3rd ed., Amsterdam: Elsevier, 2012. 157–172.
- [27] Tsoumakas G, Spyromitros-Xioufis E, Vilcek J, Vlahavas I. MULAN: A java library for multi-label learning. The Journal of Machine Learning Research, 2011, 12: 2411–2414. [doi: [10.5555/1953048.2021078](https://doi.org/10.5555/1953048.2021078)]

#### 附中文参考文献:

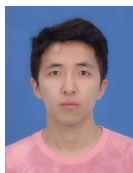
- [18] 王少博, 李宇峰. 用于多标记学习的分类器圈方法. 软件学报, 2015, 26(11): 2811–2819. <http://www.jos.org.cn/1000-9825/4908.htm> [doi: [10.13328/j.cnki.jos.004908](https://doi.org/10.13328/j.cnki.jos.004908)]
- [19] 李娜, 潘志松, 周星宇. 基于多标记重要性排序的分类器链算法. 模式识别与人工智能, 2016, 29(6): 567–575. [doi: [10.16451/j.cnki.issn1003-6059.201606011](https://doi.org/10.16451/j.cnki.issn1003-6059.201606011)]
- [21] 胡天磊, 王皓波, 尹文栋. 基于深度双向分类器链的多标签新闻分类算法. 浙江大学学报(工学版), 2019, 53(11): 2110–2117. [doi: [10.3785/j.issn.1008-973X.2019.11.008](https://doi.org/10.3785/j.issn.1008-973X.2019.11.008)]



丁家满(1974—), 男, 教授, CCF 专业会员, 主要研究领域为数据挖掘, 大数据与云计算, 软件工程.



付晓东(1975—), 男, 博士, 博士生导师, CCF 高级会员, 主要研究领域为服务计算, 智能决策, 大数据, 软件工程.



周蜀杰(1996—), 男, 硕士生, 主要研究领域为机器学习, 数据挖掘.



贾连印(1978—), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为数据库, 数据挖掘, 信息检索, 并行计算.



李润鑫(1983—), 男, 博士, CCF 专业会员, 主要研究领域为机器学习, 数据挖掘, 最优化算法.