基于 ReliefF 剪枝的多标记分类算法

刘海洋 王志海 张志东

(北京交通大学计算机与信息技术学院 北京 100044)

摘 要 多标记分类问题需要为每个实例分配多个标记.常见的多标记分类方法主要分为算法转换法和问题转换法两类.合理利用标记间的依赖关系是提升多标记分类性能的关键.在该文中,作者从不同的问题转化方法的角度,将标记间依赖关系的利用方法分为标记分组法和属性空间扩展法两种.作者发现,对于属性空间扩展法,普遍存在的难题在于如何对标记间的依赖关系进行准确度量,并选择合适的标记集合加入到属性空间中.在此基础上,作者提出了一种基于 ReliefF 剪枝的多标记分类算法(ReliefF based Stacking,RFS).算法从属性选择的角度,利用 ReliefF 方法对标记间的依赖关系进行度量,进而选择依赖关系较强的标记加入到原始属性空间中.在 9 个多标记基准数据集上的实验结果显示,RFS 算法相较于当下流行的多标记分类算法具有较为明显的优势.

关键词 多标记分类;标记间依赖关系;属性选择;ReliefF;Stacking 算法

中图法分类号 TP301 **DOI 号** 10.11897/SP.J.1016.2019.00483

ReliefF Based Pruning Model for Multi-Label Classification

LIU Hai-Yang WANG Zhi-Hai ZHANG Zhi-Dong

(*School of Computer and Information Technology, Beijing Jiaotong University, Beijing* 100044)

Abstract Multi-label classification (MLC) is a machine learning problem in which models are sought that assign a subset of labels to each instance. MLC is receiving increased attention and is relevant to many domains such as text categorization, classification of music and videos, semantic annotation of images and many more. Recently, many studies are looking for efficient and accurate algorithms to cope with multi-label classification challenge. They are usually partitioned into two main categories: algorithm adaptation and problem transformation. In multi-label classification problem the labels will not occur independent of each other; instead, there are statistical dependencies between them. Nowadays, it is commonly accepted that exploiting dependencies between the labels is the key of improving the performance of multi-label classification problem. In this paper, we divide the utilizing methods of label dependency into two groups from the perspective of different ways of problem transformation: label grouping model and feature space extending model. Label grouping model normally groups labels into several label subsets based on certain strategies or criteria to incorporate label dependences. While feature space extending model usually extends the feature space of the binary classifiers to let them discover existing label dependence by themselves. We find out that the common difficulty for both kinds of models is how to accurately measure the dependences between labels. In particular, for feature space extending model, how to choose proper labels to extend the original feature space is the key to improve classification performance. On the basis of this, we propose a ReliefF based pruning model for

收稿日期:2016-10-01;在线出版日期:2018-01-19.本课题得到国家自然科学基金(61672086,61702030,61771058)及北京市自然科学基金(4182052)资助.**刘海洋**,男,1987 年生,博士研究生,主要研究方向为多标记分类、时间序列分类. E-mail: haiyangliu@bjtu.edu.cn. **王志海**(通信作者),男,1963 年生,博士,教授,中国计算机学会(CCF)会员,主要研究领域为机器学习、数据挖掘. E-mail: zhhwang@bjtu.edu.cn. **张志东**,男,1992 年生,硕士研究生,主要研究方向为多标记分类、时间序列分类.

multi-label classification (ReliefF based Stacking, RFS). RFS measures the dependencies between labels in a feature selection perspective, and then selects the more relative labels into the original feature space. And we use a stacking based algorithm during training and prediction. The key contribution of this algorithm is threefold: (1) It provides a new method to measure the dependences between labels. Unlike existing methods measuring pair-wise label dependences, our method related to the ReliefF algorithm takes into account the effect of all interacting labels. (2) Instead of extending the original feature space with all labels, we choose the closely related labels. Thus, we can reduce noise in the data and avoid adverse effects caused by irrelevant labels. (3) In the feature selection phase, we design a brand new strategy that treats original features and label features as the same features and select together. Our empirical study is divided into two parts: a systematic study on parameters of our algorithm and a comparative study between our proposal and other multi-label classification algorithms. The effects of parameters, feature selection strategies and base classifiers on RFS are discussed in the first part of experiments. In the second part, experiment results based on 6 evaluating measures on 9 multi-label benchmark datasets show that RFS is more effective compared to other advanced multi-label classification algorithms.

Keywords multi-label classification; label dependence; feature selection; ReliefF; Stacking

1 引 言

传统的分类问题中,我们需要为实例预测单个标记的值.而在多标记分类(Multi-Label Classification, MLC)问题中,每个实例都有可能被分配到多个标记[1].近年来,多标记分类问题正受到越来越多的关注,并在多个领域得到了广泛应用,例如文本分类问题[2]、音乐[3]和视频[4]分类问题、图片语义标注问题[5]等等.

目前,为了高效并准确地解决多标记分类问题,已经有很多算法被提出.通常,它们被分为两大类.问题转化(problem transformation)方法将多标记分类问题转化为一个或多个单标记分类问题.算法转换(algorithm adaptation)方法扩展传统的单标记学习方法,使之能够直接处理多标记数据.本文我们主要探讨的是问题转化方法,而问题转化方法又主要分为二值相关法(Binary Relevance methods, BR)和标记幂集合法(Label Power-set method, LP)两种.

在目前的多标记分类学习研究中,合理地利用标记间存在的依赖关系可以带来更加理想的预测性能.通常标记不会以相互独立的形式出现,相反的,标记间往往存在着一定程度上的依赖关系.例如,当一部电影被标有"武侠"和"古装"两个标记时,它被标记为"动作"的可能性就很高;当一则新闻被标记为"军事"时,它就不太可能同时拥有"娱乐"标记.从学习和预测的观点来看,标记间的依赖关系是重要的信息来源,可以用于弥补待预测实例本身信息不足的缺陷.因此,近年来大量多标记学习的研究工作的重点都在于标记间依赖关系的发现和利用.

Zhang 等人中根据对标记间相关性的不同处理方式将现有标记关系发现策略分为三类:"一阶",即逐一学习单个标记而忽略标记之间的相关性;"二阶",即学习两两标记之间(pair-wise)的相关性;"高阶",即学习标记间的高阶相关性[6].在另一篇文献中,作者对标记间的依赖关系给出了理论分析,从统计的角度将标记间的依赖关系分为条件(conditional)依赖关系和非条件(unconditional)依赖关系两种[7].

本文中,我们从不同的问题转化方法的角度,对标记间依赖关系的利用方法进行了区分.我们认为,标记间依赖关系的利用方法可以分为标记分组法和属性空间扩展法两种.在之后的章节中,我们分别对这两种利用方法进行了举例说明,并重点对属性空间扩展法现阶段研究中存在的问题做出了具体分析.从中我们发现,不论是标记分组法还是属性空间扩展法,他们存在的共同难题都在于如何对标记间的依赖关系进行准确度量.特别的,对于属性空间扩展法,如何选择合适的标记集合加入到属性空间中是提升分类精度的关键.在此基础上,我们提出了一种基于 ReliefF 剪枝的多标记分类算法.

本文算法的贡献主要在于：(1) 我们提供了一种标记间依赖关系的度量方法. 不同于已有的方法，我们的度量方法在计算标记间依赖关系时，不仅考虑到了标记间的两两关系，还考虑了其他标记的影响；(2) 在对原始属性空间进行扩展的过程中，相对于加入全部标记，我们的策略是有选择性的将与当前待分标记密切相关的标记加入到属性空间中，这样可以减少数据中的噪声，避免那些不相关的标记对分类造成不良影响；(3) 在属性选择过程中，我们提供了一种全新的策略，即将原始属性和后加入进来的标记属性视为同样的属性，共同进行属性选择.

本文在第 2 节中，将依据标记间依赖关系的利用方法对近年来的多标记分类方法进行分类综述分析；在此基础上，我们将在第 3 节中具体阐述本文算法的设计思路与实现过程；相关实验的结果与分析将在第 4 节中给出；最后，我们在第 5 节中对本文的工作进行总结与展望.

2　标记间依赖关系的利用

正如我们在上文中提到的，标记间的依赖关系普遍存在于多标记数据中，是多标记学习中的重要信息，如何有效利用标记间的依赖关系是提高分类性能的关键. 通常，每当一种新的标记间依赖关系利用方法被发现，就会伴随着一种新的多标记分类算法被提出. 在本节中，我们将回顾近年来出现的多标记分类算法. 根据它们利用标记间依赖关系方法的不同，我们将这些算法分为两类：标记分组法与属性空间扩展法.

在具体研究标记间依赖关系的利用问题之前，我们在这里给出多标记学习问题的符号表示方法. 我们定义标记空间为 $\mathcal{Y}=\{0,1\}^m$，标记集合 $\boldsymbol{y}=\{y_1,y_2,\cdots,y_m\}\in\mathcal{Y}$；属性空间定义为 $\mathcal{X}$，属性集合用 $\boldsymbol{x}=\{x_1,x_2,\cdots,x_d\}\in\mathbb{R}^d$ 表示. 这样，多标记数据集就可以表示为 $D=\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\mid i=1,2,\cdots,n\}\in(\mathcal{X}\times\mathcal{Y})$. 实例 $\boldsymbol{x}_i$ 的相关标记集合 $\boldsymbol{y}_i$ 还可以表示成一个长度为 m 的 0/1 向量 $\boldsymbol{c}_i=(c_{i1},c_{i2},\cdots,c_{im})$，其中 c_{ij} 对应标记空间中的第 j 个标记 y_j，$c_{ij}=1$ 表示 $y_j\in\boldsymbol{y}_i$，即 y_j 为 $\boldsymbol{x}_i$ 的相关标记，而 $c_{ij}=0$ 则表示 $y_j\notin\boldsymbol{y}_i$，即 y_j 为 $\boldsymbol{x}_i$ 的非相关标记. 解决多标记分类问题就是要在训练集合 D 上学习一个分类器(映射) $h:\mathcal{X}\to\mathcal{Y}$，对任意给定未知标记的实例 $\boldsymbol{x}$，分类器 h 对 $\boldsymbol{x}$ 的相关标记集合做出预测，即 $\hat{\boldsymbol{y}}=h(\boldsymbol{x})$.

2.1　标记分组法

在传统的 LP 算法中，针对多标记数据集 $D=\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\mid i=1,2,\cdots,n\}$，该方法直接将各实例 $\boldsymbol{x}_i$ 对应的 $\boldsymbol{y}_i$ 整体看作一个新标记，即新的标记集合由原标记空间 $\boldsymbol{y}=\{y_1,y_2,\cdots,y_m\}$ 的所有子集构成，共有 2^m 个可能的值. 在训练过程中，LP 算法将实例对应的多个标记看作整体进行学习，通过这样的方式，LP 算法潜在地利用了标记间的依赖关系. 然而不同的标记组合的个数随 m 呈指数级增长. 同时，每种标记组合下的实例个数可能非常稀少，这也是 LP 算法主要的缺陷之一.

为了在利用标记间依赖关系的同时避免传统 LP 算法的缺陷，标记分组法是研究人员普遍会使用的方法. 这类方法通常首先以某种策略或标准将原始标记集合进行分组，进而在每组标记内部分别构建分类器，这样每组中标记组合的个数将远远小于原始标记个数，最终新实例的分类结果由各个分类器的分类结果集成得到. 这类方法中的代表算法是 RAkEL 算法[8].

RAkEL(RAndom k-labELsets)算法将原始标记集合随机分割成 q 个大小为 k 的标记集合，为每个标记集合建立一个 LP 分类器，最终的分类结果由各个 LP 分类器的分类结果集成而成. 这样，当取 $k\ll m$ 时，相对于传统的 LP 算法，RAkEL 的计算复杂度可以得到显著降低，LP 中实例分布严重不均的问题也得到了缓解. 然而正如文中所提到的，RAkEL 在标记选择上的随机特性有可能会对分类性能造成不好的影响. 针对这样的问题，Rokach 等人试图使用某种特定的子集选择方法来替代 RAkEL 算法中的随机选择方法[9]. 理想的情况是，在覆盖所有标记的前提下，选择最少数量的子集并取得最好的预测效果. 作者从集合覆盖问题(Set Covering Problem，SCP)的角度研究子集选择问题，并采取了一种贪婪近似算法来解决这一问题.

为解决 LP 方法中的数据稀疏问题，Read 等人提出了 PS(Pruned Sets)和 EPS(Ensemble of Pruned Sets)方法[10]. 对于标记间依赖关系的利用，作者认为应该重点抓住那些最为关键的依赖关系，通过对出现频率较低的标记组合进行剪枝，可以忽略掉那些不重要的依赖关系. 在之后的步骤中，一些被剪枝掉的实例会经过标记集合分解被重新加回到数据集中，最后在新生成的数据集上构建 LP 分类器. PS 和 EPS 算法存在着一些局限性：首先，当数据集中存在大量不同的标记组合，且实例在这些标记组合上的分布相对较为均匀时，算法的有效性会受到严重影响；另外，PS 和 EPS 算法在剪枝掉一部分训练实例时必然会造成一定程度上的信息损失，

同时还会加回一些标记集合被分解后的实例，算法需要在这二者之间进行权衡. 这就需要在应用算法之前选取一些非平凡参数值，或是执行交叉验证来调整参数.

2.2 属性空间扩展法

从概率的角度出发，多标记学习也可以看作求多个标记的联合条件概率 $P(\boldsymbol{y}|\boldsymbol{x})$ 的问题. 此时，为实例 $\boldsymbol{x}$ 预测的最优标记向量 $\boldsymbol{y}^*$ 应该是能使联合条件概率最大的那个向量，如式(1)所示.

$$\boldsymbol{y}^*=\arg\max_{\boldsymbol{y}\subseteq\{0,1\}^m} P(\boldsymbol{y}|\boldsymbol{x}) \tag{1}$$

与 LP 类算法不同，BR 算法首先假设标记间相互独立，各个标记的学习过程也相互独立，仅依赖于标记各自的原始属性(如式(2)所示)，即为每个标记单独建立一个单标记二值分类器，最终的分类结果由各个单标记分类器的分类结果组合而成. 从而无法利用标记间的依赖关系. 图 1 展示了一个标记集合为 $\boldsymbol{y}=\{y_1,y_2,y_3\}$ 的数据利用 BR 算法进行分类的过程. 其中 $\boldsymbol{x}_i$ 为待预测实例，最终 $\{\hat{y}_1,\hat{y}_2,\hat{y}_3\}$ 为 BR 分类器为 $\boldsymbol{x}_i$ 预测的标记集合.

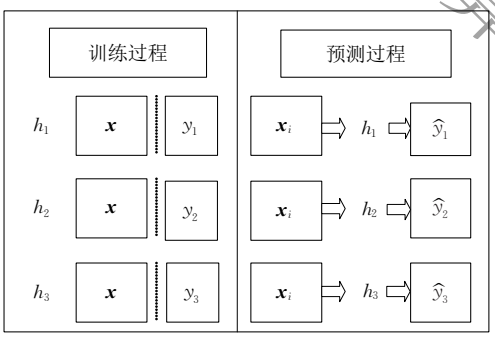

图 1 $\boldsymbol{y}=\{y_1,y_2,y_3\}$ 时 BR 分类器分类举例

$$P(\boldsymbol{y}|\boldsymbol{x})=\prod_{j=1}^{m} P(y_j|\boldsymbol{x}) \tag{2}$$

BR 算法的优势在于其易于实现，且计算复杂度低. 为弥补 BR 算法无法考虑标记间依赖关系的缺陷，研究人员提出了多种基于 BR 算法的改进方法，这类算法一般是通过扩充原始属性空间的办法来达到这样的目的. 改进的 BR 算法在预测标记 y_i 的过程中不仅依赖于原始特征向量 $\boldsymbol{x}$，还依赖于某些其它标记.

其中一类算法将标记间的链式依赖结构应用于 BR 算法，这一类算法的代表是分类器链算法(Classifier Chains，CC)[11]. 图 2 中展示了当标记顺序为 $y_1\prec y_2\prec y_3$ 时分类器链算法的分类过程. 与 BR 算法类似，分类器链算法为每一个标记建立一个二值分类器，所有这些分类器被连接成为一个有序链. 链中每个分类器使用在其之前所有分类器的分类结果来扩展其特征空间. 也就是说，链中某一标记的分类结果是由原始特征加上之前标记的预测结果共同决定的(如式(3)所示). 这样一来，在分类过程中就考虑到了标记间的依赖关系，避免了 BR 算法的标记独立性假设. 文中实验显示，在绝大多数数据集中，CC 提升了 BR 算法的分类准确率. 分类器链算法的不足在于链顺序本身会对准确率造成影响，链中某个效果欠佳的分类器会给在其后面的分类器带来误差. 采取启发式策略选择分类器链的顺序或构造集成分类器链可以解决上述问题，然而这两种措施都会大大增加算法的计算时间.

$$P(\boldsymbol{y}|\boldsymbol{x})=\prod_{j=1}^{m} P(y_j|\boldsymbol{x},y_1,\cdots,y_{j-1}) \tag{3}$$

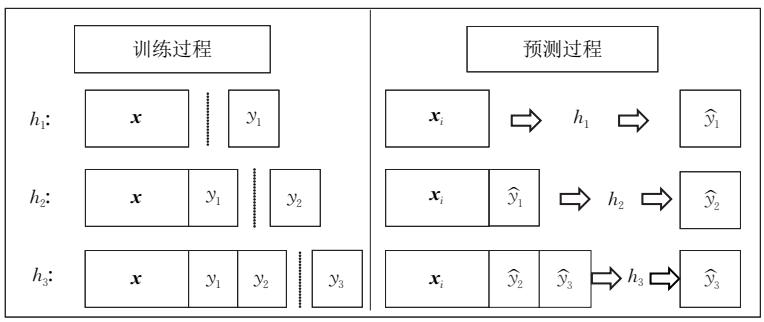

图 2 当标记顺序为 $y_1\prec y_2\prec y_3$ 时 CC 分类器分类举例

在分类器链算法提出之后，出现了一系列针对 CC 的改进算法. 其中，Dembczynski 等人提出了一种概率分类器链算法(Probabilistic Classifier Chains，PCC)[12]. 作者使用概率的乘法准则来计算每种标记组合的条件概率. 为了估计标记的联合分布，PCC 为每个标记学习一个模型，最终的分类结果由计算得到的联合分布得出. 文中从理论上和实验中证明了 PCC 可以比原始的分类器链算法有更好的预测表现，然而代价是比 CC 高的多的计算复杂度. 事实上，PCC 也只能处理标记个数较少(不多于 15 个)的数据集合. Senge 等人在训练阶段使用预测的标记值代替分类器链算法中使用的真实标记值[13]. 这

样做的动机在于预测值与真实值往往是不同分布的，真实值作为分类器中额外添加的特征，有可能会成为噪音特征，不仅难以获取标记间真实的依赖关系，还有可能对分类效果带来不好的影响.

将堆栈泛化(stacked generalization，简称 Stacking)方法应用于多标记分类场景，是另一种在 BR 方法中加入标记间依赖关系信息的策略. 这种方法是由 Godbole 和 Sharawagi 首次提出的[14]. 在训练阶段，他们建立两组 BR 分类器. 第一组(first-level 或 base-level)分类器为传统的 BR 分类器，第二组(meta-level)分类器将第一组分类器的输出结果作为输入，用以扩展其特征空间. 这样，Stacking 方法中每个标记的预测都依赖于原始属性与全部标记(如式(4)所示). 除去与标记个数相关的线性复杂度，Stacking 方法保留了我们前文提到的 BR 方法的全部优势. 然而，据我们所知，有些时候某些标记间是完全不存在联系的，Stacking 方法笼统的将所有标记都加入到 meta-level 分类模型的特征空间中. 模型中不相关的标记不仅缺乏对分类有利的信息，而且还会带来 base-level 中固有存在的噪声信息.

$$P(\boldsymbol{y}|\boldsymbol{x})=\prod_{j=1}^{m}P(y_j|\boldsymbol{x},y_1,\cdots,y_m) \tag{4}$$

为了应对这样的问题，Tsoumakas 等人提出了一种针对 Stacking 算法的剪枝算法(Correlation-Based Pruning of Stacking，CPS)[15]. 该算法试图在建立模式之前通过计算标记间的相关性系数 phi，来量化标记间的两两依赖程度，并以此对 base-level 中的标记进行筛选，仅选择与当前标记最相关的若干个标记加入到 meta-level 的特征空间中. 这样的剪枝过程消减了 meta-level 分类模型中特征空间的维度，因而提升了算法的效率. 并且，根据文中给出的结果，在某些数据集上可以带来预测精度的提升. 尽管该算法考虑到了标记间的依赖关系，并对依赖关系进行了定量计算，却仅仅在 meta-level 的模型中利用了这样的依赖关系.

BR＋算法是一种与 Stacking 算法类似的算法[16]. BR＋与 Stacking 算法有两个主要的区别：(1) Stacking 使用 SVM 作为基分类器，而 BR＋算法不局限于使用某种特定的基分类器；(2) Stacking 算法在 meta-level 阶段使用所有标记在 base-level 阶段的预测值，而 BR＋算法会从中剔除被预测标记本身，只保留其他标记的预测值(如式(5)所示). 作者认为这样可以避免过拟合现象的出现. 图 3 中展示了 BR＋算法的分类过程. DBR(Dependent Binary Relevance)算法同样以 Stacking 方法为基础，算法中重点探讨了在训练过程中使用真实标记值和预测值的优劣势[17].

$$P(\boldsymbol{y}|\boldsymbol{x})=\prod_{j=1}^{m}P(y_j|\boldsymbol{x},y_1,\cdots,y_{j-1},y_{j+1},\cdots,y_m) \tag{5}$$

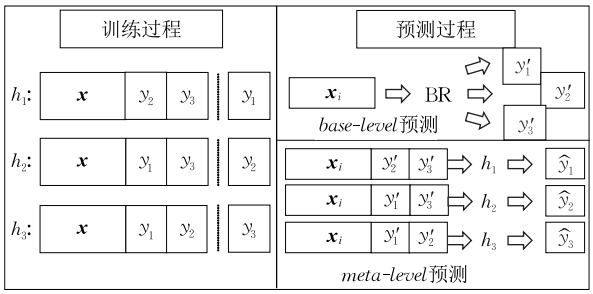

图 3　当 $\boldsymbol{y}=\{y_1,y_2,y_3\}$ 时 BR＋分类器分类举例

综上所述，我们可以看出目前 BR 类算法在对于标记间依赖关系的利用中存在如下几点问题：

(1) 无论是链式算法还是堆栈泛化类算法，往往都是笼统的将部分或全部标记加入到数据的原始属性中，并没有定量的对标记间的依赖关系进行计算.

(2) 链式算法只选择了部分标记间的依赖关系，而堆栈泛化类算法虽然考虑了完全的依赖关系，却存在大量冗余，如何针对每一标记准确选择具有依赖关系的标记集合加入到属性空间中是问题的关键.

(3) 既然我们将标记加入到了属性空间中，那么筛选标记的过程实际就看作为单标记下的属性选择问题，我们可以将原始属性和后加入的标记属性同等对待，以同样的标准来衡量它们的重要性. 而现有的筛选算法的筛选对象仅仅是后加入到属性空间中的标记属性，并且其使用的评价标准如卡方系数无法考虑到属性间的相互作用.

针对以上问题，我们提出了一种基于 ReliefF 剪枝的多标记分类算法. 我们的算法将所有的原始属性和标记一同视为分类中的条件属性，以定性的考虑标记间的依赖关系. 在接下来的属性选择步骤中，我们采取了两种策略，分别对标记和标记与原始属性的整体进行属性选择操作. 不管是哪一种策略，我们都对标记间的依赖关系进行了定量计算，作为筛选的依据. 在计算两个标记间依赖关系的同时，我们还考虑了其他标记的影响. 在本文第 3 节中，我们将详细介绍我们算法的实现过程.

2.3　多标记属性选择方法

属性选择(Feature Selection，FS)是机器学习

和数据挖掘中的重要任务，它可以通过移除数据中的无关以及冗余属性来有效降低数据集合的维度，加快算法的运算速度，提升运算的性能. 然而，目前大多数的属性选择方法研究是针对单标记分类场景的，根据 Spolaôr 等人对多标属性选择方法研究给出的系统性综述[18]，这一领域的相关文献还很少.

与单标记属性选择方法类似，多标记属性选择方法也分为过滤式(filter)、封装式(wrapper)、嵌入式(embedded)三类. 其中过滤式方法不仅具有快速且易于实现的优点，而且这种方法在过滤属性的过程中是独立于具体的学习算法的，它仅使用数据的一般特性来选择某些属性并剔除其他属性.

过滤式方法需要我们对所有属性的重要程度进行度量，来比较它们在分类过程中的贡献大小. 常用的评价指标有文献[15]中使用的 phi 系数，以及卡方值(Chi-square)、信息增益(information gain)等. 这里我们主要介绍 ReliefF 方法[19]，相对于其他的指标在计算时假设属性间相互独立，ReliefF 的主要优势在于它考虑到了属性间的相互关系.

在文献[18]中，作者给出了 ReliefF 算法在多标记分类场景中的应用，提出了 RF-BR 和 RF-LP 算法. RF-BR 算法首先利用 BR 方法将多标记转换为 m 个二值单标记数据，进而使用 ReliefF 算法对每个数据中的属性集合进行评价，得到 m 组属性权重向量. 在接下来的步骤中，作者对每个属性在各组数据中的权重取均值，结果大于阈值的属性将被最终选择. RF-LP 算法与之类似，只不过在问题转换步骤中使用的是 LP 方法. 在之后的一篇文章中，Spolaôr 等人进一步对基于 ReliefF 和基于信息增益的多标记属性选择方法进行了实验评价，验证了 ReliefF 评价方法相对于信息增益在多标记场景中更具优势[20]. Reyes 等人提出了三种 ReliefF 方法在多标记场景中的应用算法[21]. 其中 PPT-ReliefF 需要将多标记分类问题转换为单标记分类问题，而 ReliefF-ML 和 RReliefF-ML 算法则是通过调整原始的 ReliefF 算法使之能够直接应用于多标记分类场景. 除此之外，国内研究人员对于 ReliefF 方法的研究也取得了一定成就，Wang 等人在传统 ReliefF 方法中加入了冗余分析，提出了使用均值进行随机选择的方法，提升了 ReliefF 方法的抗波动性，并通过实验验证了改进算法在高分辨率遥感影像分类中的有效性[22].

在我们目前所能找到的多标记属性选择算法的文献中，所有算法都仅仅是对数据集的原始属性进行选择或加权. 在本文中，我们会从一个全新的角度看待多标记属性选择方法. 我们的算法将标记也作为属性的一部分加入到原始的属性空间中，在此基础上使用属性选择方法对标记(或原始属性加标记)进行评价. 这样的方法不仅可以在分类过程中定性的考虑到标记间的依赖关系，还可以借助 ReliefF 这样的单属性评价方法定量的去计算标记间的依赖关系；既是对标记间依赖关系利用方法的深入扩展，也是多标记属性选择方法的全新应用.

3 算法实现

从前文的总结中我们可以发现，合理利用标记间依赖关系的关键在于以下两点：(1) 如何准确度量标记间的依赖关系；(2) 如何为每个标记选择具有依赖关系的标记子集. 针对这样的问题，我们设计了一种基于 ReliefF 剪枝的多标记分类算法，算法借助 ReliefF 算法对标记间的依赖关系进行定量计算，并以此为标准对每个标记所依赖的标记进行选择. 在本节中我们将对我们提出的算法给出具体介绍.

3.1 问题转化步骤

当面对一个多标记分类问题时，我们首先对数据集进行转化，将多标记分类问题转化为单标记分类问题. 这里，为了利用多标记间的依赖关系，在进行传统的 BR 转换之后，我们将除当前标记以外的所有其他标记加入到数据原始属性空间中. 也就是说，每一个实例的属性都由初始的 $\boldsymbol{x}$ 加上了 $m-1$ 个二值属性.

这样，原始数据集合 $D=\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\mid i=1,\cdots,n\}$ 就转化成了式(6)中的数据集 S.

$$S=\{((\boldsymbol{x}_i,y_{i,1},y_{i,2},\cdots,y_{i,j-1},y_{i,j+1},\cdots,y_{i,m}),y_{i,j})\mid i=1,\cdots,n,j=1,\cdots,m\} \tag{6}$$

数据集中的实例个数由原始的 n 个变为了 $m\times n$ 个，其中每个实例的标记由原始的 m 个变为了 1 个，属性个数由原始的 d 个变为了 $d+m-1$ 个. 我们需要为每一个标记 y_j 训练一个分类器 $h_j(\boldsymbol{x},y_1,\cdots,y_{j-1},y_{j+1},\cdots,y_m)\rightarrow y_j$，也就是说总共需要训练 m 个分类器.

3.2 属性选择步骤

经过上一步骤的问题转化过程，我们得到了一组属性空间扩展后的二值单标记数据，新的数据集

合的属性空间中包括原始属性以及除当前标记本身以外的所有其他标记属性. 以往的 stacking 类算法(例如 BR+,DBR)将直接在这样的数据上训练分类器. 这样的方法假设所有原始属性和标记属性对分类的贡献相同,然而这样的假设显然是不符合实际应用情况的. 接下来我们使用 ReliefF 算法对这些属性与当前标记的依赖程度进行度量,筛选出对分类贡献大的属性.

我们选择 ReliefF 度量标记间依赖关系的原因有以下两点:首先,ReliefF 作为一种属性选择方法,对属性间的相互作用敏感,能够发现属性间的高阶相关性,善于处理属性间具有强相关性的数据. 这样的特性与我们的多标记数据的特点恰好吻合,即标记间普遍存在着依赖关系. 使用 ReliefF 度量两个标记的依赖程度时,还可以考虑到其他标记的影响. 其次,ReliefF 算法在多标记分类场景中的有效性已经得到了实验验证,例如 RF-BR 和 RF-LP. 它们分别是基于 BR 和 LP 转化的,且算法中待选择的属性只是数据中的原始属性. 这里,我们使用的是不同的问题转化方法,并且我们需要进行选择的属性还包括除当前标记以外的其他标记属性.

设标记 $y_j(j=1,\cdots,m)$ 所对应的分类器 $h_j(\boldsymbol{x},y_1,\cdots,y_{j-1},y_{j+1},\cdots,y_m)$ 经属性选择操作后,最终得到的属性集合为 $Pa(y_j)$,其中原始属性集合为 $Pa_{\boldsymbol{x}}(y_j)$,依赖标记集合为 $Pa_{\boldsymbol{y}}(y_j)$,三者之间关系满足式(7).

$$Pa(y_j)=Pa_{\boldsymbol{x}}(y_j)\bigcup Pa_{\boldsymbol{y}}(y_j) \tag{7}$$

设属性选择百分比参数为 t,针对问题转化后的数据特性,我们设计了 3 种不同的属性选择策略.

(1) RFS(ReliefF based Stacking). 保留全部原始属性,即 $Pa_{\boldsymbol{x}}(y_j)=\boldsymbol{x}$,只对标记属性进行评价并排序,保留的标记属性个数 $|Pa_{\boldsymbol{y}}(y_j)|=[t\times m]$,最终属性集合 $Pa(y_j)=\boldsymbol{x}\bigcup Pa_{\boldsymbol{y}}(y_j)$.

(2) RFS-S(RFS-Separated). 对全部原始属性和标记属性进行评价,并分别进行排序,保留前 $[t\times d]$ 个原始属性和前 $[t\times m]$ 个标记属性分别构成 $Pa_{\boldsymbol{x}}(y_j)$ 和 $Pa_{\boldsymbol{y}}(y_j)$,最终属性集合 $Pa(y_j)$ 由二者取并集得到.

(3) RFS-J(RFS-Joint). 对全部原始属性和标记属性进行评价,并共同进行排序,保留前 $[t\times(d+m)]$ 个属性,直接得到最终属性集合 $Pa(y_j)$.

ReliefF 方法根据属性值对于近邻实例的区分能力来评价属性,它的具体思想是:如果一个属性在两个标记不同的近邻实例上取不同值,则奖励该属性;如果一个属性在两个标记相同的近邻实例上取不同值,则惩罚该属性. 对于每个属性,ReliefF 输出一个值 w,w 值的大小从 -1 到 1,有着正值 w 的属性的 w 值越大表明该属性的重要程度越高.

设 S_a 和 S_b 是我们的训练集合中两个具有相同标记 y_j 的实例,它们的标记值分别是 c_a 和 c_b. 这里我们首先给出属性 $p(p\in\boldsymbol{x}\bigcup\boldsymbol{y})$ 在两个实例 S_a 和 S_b 上的差的定义:

如果 p 是数值型属性,则

$$diff(p,S_a,S_b)=\left|\frac{c_a-c_b}{\max_p-\min_p}\right| \tag{8}$$

其中 $\max_p$ 和 $\min_p$ 分别是属性 p 在训练集中的最大值和最小值.

如果 p 是名称型属性,则

$$diff(p,S_a,S_b)=\begin{cases}0, & c_a=c_b\\ 1, & c_a\neq c_b\end{cases} \tag{9}$$

根据属性在两个实例间差的定义,我们将两个实例间的距离 $dis(S_a,S_b)$ 定义为所有属性在两实例间差的和,如式(10)所示:

$$dis(S_a,S_b)=\sum_{p}^{p\in(\boldsymbol{x}\bigcup\boldsymbol{y})}diff(p,S_a,S_b) \tag{10}$$

在对标记 $y_j(j=1,\cdots,m)$ 进行属性选择的过程中,首先我们将所有属性的权重 w_p 初始化为 0,并找到所有标记为 y_j 的实例集合. 接着在该集合中随机选取一实例 S_i,分别从标记值与 S_i 相同和不同的实例中找到一个与 S_i 距离最小的实例,分别用 $Hit(E_i)$ 和 $Miss(E_i)$ 表示. 最后利用式(11)来更新每个属性的权重 w_p.

$$w_p=w_p-diff(p,S_i,Hit(S_i))/r+diff(p,S_i,Miss(S_i))/r \tag{11}$$

为避免一次抽样的随机性,上述过程需要迭代 r 次. 根据 ReliefF 算法的思想,对类值区分能力较强的属性 p 应该表现为在异类间差异较大而同类间差异较小,反映在式(11)上即为 $diff(p,S_i,Miss(S_i))$ 值较大,$diff(p,S_i,Hit(S_i))$ 值较小. 我们以策略(1)的 RFS 算法为例,上述过程的伪代码表示如算法 1 所示.

算法 1.　RFS 算法.

输入:问题转化后的训练集合 S,迭代次数 r,属性选择百分比 t

输出:各标记的依赖属性集合 $Pa(y_j),j=1,\cdots,m$

```
BEGIN
  FOR j←1 to m DO
```

FOR all $p\in(\boldsymbol{y}-\{y_j\})$ DO
$w_p=0$
FOR $i\leftarrow 1$ to r DO
在标记为 y_j 的实例中随机选取一个实例 S_i
在与 S_i 类值相同的实例中找到最近邻实例 $Hit(S_i)$
在与 S_i 类值不同的实例中找到最近邻实例 $Miss(S_i)$
$w_p\leftarrow w_p-diff(p,S_i,Hit(S_i))/r+diff(p,S_i,Miss(S_i))/r$
END FOR
END FOR
对 w 从大到小排序,取前$[t\times m]$个属性构成 $Pa_y(y_j)$
$Pa(y_j)\leftarrow\boldsymbol{x}\cup Pa_y(y_j)$
END FOR
END

3.3 训练与测试

在得到问题转化后的训练集以及各标记的依赖属性集合 $Pa(y_j)$之后,我们就可以进行分类模型的训练了.我们需要为每一个标记训练一个单独的分类器,最终得到一组 m 个二值单标记分类器.

当我们需要对一个实例进行预测时,集合 $Pa(y_j)$中原始属性部分 $Pa_x(y_j)$是已知的,而标记属性部分 $Pa_y(y_j)$是未知的.这里我们首先在原始数据集上用 BR 算法对 $Pa_y(y_j)$中的标记值进行预测,参照 stacking 算法中的命名方式,我们称这一过程为 base-level 预测.在此之后,我们利用式(12)中的分类器对实例的各个标记值进行预测,得到 $\hat{y}_j(j=1,\cdots,m)$,我们称这一过程为 meta-level 预测.最终待预测实例的预测结果为 $\hat{\boldsymbol{y}}=\{\hat{y}_1,\cdots,\hat{y}_m\}$.

$$h_j(Pa(y_j))\rightarrow y_j,\ j=1,\cdots,m \tag{12}$$

4 实 验

我们的实验是在 Mulan① 平台和 Weka② 平台上进行的,Mulan 是目前广泛使用的多标记学习开源平台[1].本节将介绍我们的实验过程与实验结果,并针对实验结果进行讨论.

4.1 数据集合

在实验中,我们选择了 9 个多标记基准数据集合,这些数据集合全部来自于 Mulan 网站③.9 个数据集合涉及文本、音频、生物、视频 4 个领域的信息,它们的具体统计数据如表 1 所示.

表 1 实验数据集合描述

name	domain	instances	attributes		labels	cardinality	density	distinct
			nominal	numeric				
birds	audio	645	2	258	19	1.014	0.053	133
CAL500	music	502	0	68	174	26.044	0.150	502
emotions	music	593	0	72	6	1.869	0.311	27
enron	text	1702	1001	0	53	3.378	0.064	753
genbase	biology	662	1186	0	27	1.252	0.046	32
medical	text	978	1449	0	45	1.245	0.028	94
yeast	biology	2417	0	103	14	4.237	0.303	198

表 1 中 name、domain 分别标明数据集的名称及所来自的领域;instances、attributes(nominal/numeric)、labels 分别标明数据集的实例个数、属性个数(名称型/数值型)、标记个数.cardinality 即标记基数(Label Cardinality),用于统计训练集中实例的平均标记个数,其定义如式(13)所示.

$$\text{Label Cardinality}(D)=\frac{1}{n}\sum_{i=1}^{n}|\boldsymbol{y}_i| \tag{13}$$

density 即标记密度(Label Density),用于统计数据集中实例的平均标记密度,其定义如式(14)所示.

$$\text{Label Density}(D)=\frac{1}{n}\sum_{i=1}^{n}\left|\frac{\boldsymbol{y}_i}{m}\right| \tag{14}$$

distinct 即标记集个数(Distinct Label Set),表示的是数据集合中出现过的所有不同的标记子集的个数.

4.2 评价指标

为评价本文算法的性能,我们选择使用 6 种多标记分类评价指标,它们分别是汉明损失(*Hamming Loss*)、子集正确率(*Subset Accuracy*)、召回率(*Recall*)、精确率(*Precision*)、*F*1 测量(*F*1-*measure*)、准确率(*Accuracy*).

(1) Hamming 损失.Hamming 损失计算的是在整个测试集 T 上被预测错误的标记比例的均值.定义如式(15)所示.

$$Hamming\ Loss(h,T)=\frac{1}{n}\sum_{i=1}^{n}\frac{\boldsymbol{y}_i\oplus\hat{\boldsymbol{y}}_i}{m} \tag{15}$$

① http://mulan.sourceforge.net/
② http://www.cs.waikato.ac.nz/ml/weka/
③ http://mulan.sourceforge.net/datasets-mlc.html

其中，$\boldsymbol{y}_i$ 为第 i 个测试实例的标记向量，$\hat{\boldsymbol{y}}_i$ 为分类器 h 为其预测的标记向量. $\oplus$ 为异或符号，统计两个向量中值不相同的对应项个数. Hamming 损失的值在 [0,1] 间变化，值越大表明分类器对测试实例的多个标记分错的比例就越大.

(2) 子集正确率(*Subset Accuracy*). 子集正确率比较的是各测试实例的真实标记集合和预测标记集合，只有当两个集合相等时才认为分类正确，否则就认为分类错误. 因此，子集正确率实质上统计了被完全正确分类的实例的比例，定义如式(16)所示.

$$Subset\ Accuracy(h,T)=\frac{1}{n}\sum_{i=1}^{n}I(\boldsymbol{y}_i=\hat{\boldsymbol{y}}_i) \tag{16}$$

其中 $I(\text{true})=1$，$I(\text{false})=0$. 一般认为这是一种过于严格的评价标准，因为即使两个集合中只有一对对应元素不相等，该度量方法就会认为分类完全错误，并没有考虑到在两个集合大多数元素都相同的情况下，预测的标记集合已经很接近实例的真实标记集合.

(3) 召回率(*Recall*). 召回率统计每个测试实例真实标记集与相应的预测标记集的交集大小与真实标记集大小的比，并在所有测试实例上求均值. 其定义如式(17)所示.

$$Recall(h,T)=\frac{1}{n}\sum_{i=1}^{n}\frac{|\boldsymbol{y}_i\cap\hat{\boldsymbol{y}}_i|}{|\boldsymbol{y}_i|} \tag{17}$$

(4) 精确率(*Precision*). 精确率统计每个测试实例的真实标记集与相应的预测标记集的交集大小与预测标记集大小的比，并求均值. 其定义如式(18)所示.

$$Precison(h,T)=\frac{1}{n}\sum_{i=1}^{n}\frac{|\boldsymbol{y}_i\cap\hat{\boldsymbol{y}}_i|}{|\hat{\boldsymbol{y}}_i|} \tag{18}$$

(5) *F*1 测量(*F*1-*measure*). 精确率和召回率是两个相互制约的标准. 为了提高召回率，需要尽可能地增加预测标记个数，即预测标记集合越大越好. 而增加预测标记个数就很可能会降低精确率. 因此，为了平衡这两个标准，以能更准确全面地评价分类器性能. 人们提出了 *F*1 测量，其定义如式(19)所示.

$$F1(h,T)=\frac{2\cdot Recall(h,T)\cdot Precision(h,T)}{Recall(h,T)+Precision(h,T)} \tag{19}$$

(6) 准确率(*Accuracy*). 准确率统计每个真实标记集与预测标记集的交集大小与真实标记集与预测标记集的并集大小的比，并在实例上求均值. 其定义如式(20)所示.

$$Accuracy(h,T)=\frac{1}{n}\sum_{i=1}^{n}\left|\frac{\boldsymbol{y}_i\cap\hat{\boldsymbol{y}}_i}{\boldsymbol{y}_i\cup\hat{\boldsymbol{y}}_i}\right| \tag{20}$$

4.3　实验设置

我们的实验总体分为两大部分. 首先我们将针对我们算法的内部特性进行实验验证，包括算法参数与属性选择策略对结果的影响. 在第二部分实验中，我们将我们的算法与当下流行的算法进行横向比较. 本文中所有的实验结果都是经过 10 重交叉验证后得到的结果.

在第一部分实验中，我们首先研究属性选择百分比参数 t 对算法结果的影响. 这里我们选择 J48 决策树分类器作为 base-level 和 meta-level 的基分类器，以策略(1)RFS 算法进行属性选择，即保留全部原始属性，只对标记属性进行筛选. 参数 t 实际上就是扩展后的属性空间中标记属性占(除当前标记以外)全部标记的百分比，我们在 0.1 到 1 之间选取了 10 个 t 值在各数据集上进行测试，实验结果如图 4、图 5 所示.

这里为了节省空间，我们只给出了在 enron 和 CAL500 两个数据集上的实验结果. 从图 5 我们可以看出，在 CAL500 数据集上，各项评价指标基本上都是随着 t 值增大而变好. 究其原因，我们需要注意到这个实验中一个关键的数据统计量，即原始属性个数与标记个数的比值. CAL500 数据集的这一

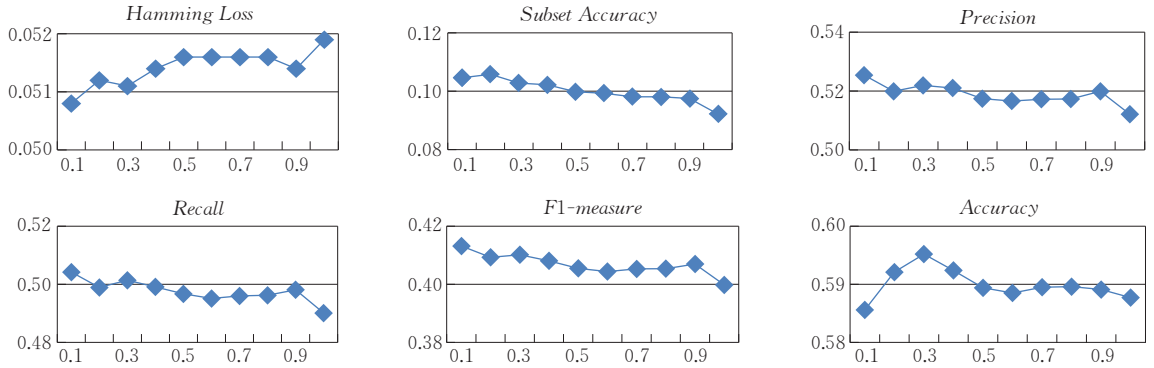

图 4　enron 数据集 t 值实验结果

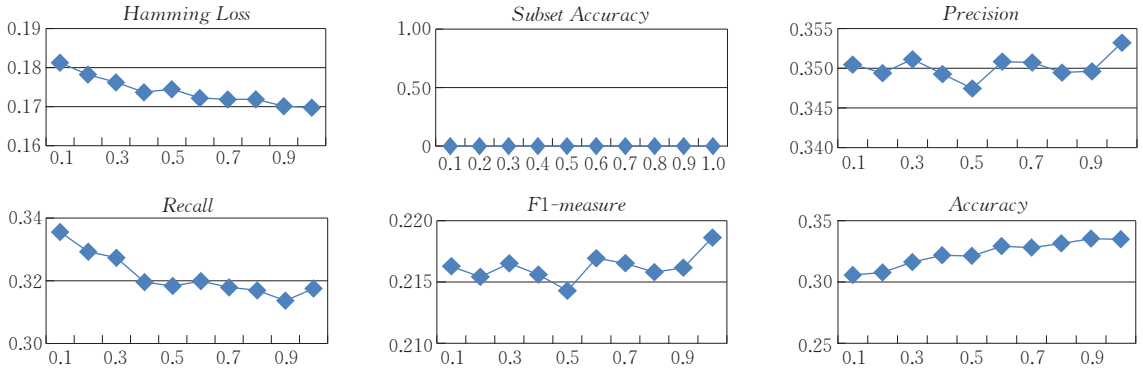

图 5 CAL500 数据集 t 值实验结果

比值为 0.39∶1,是所有 9 个数据集中唯一一个原始属性数比标记数少的数据集.因此,当加入到属性空间中的标记属性越多,就越能补充原始属性信息的不足,实验效果也就越好.反观 enron 数据集,它的属性个数与标记个数的比值是 18.89∶1.图 4 的实验结果显示,当 t 值取 0.2 或 0.3 时,各项评价指标便可达到一个较为理想的值.这说明为每个标记选择合适的依赖标记加入到原始属性空间中,可以提升算法的性能,然而加入过少或过多的标记都难以达到理想的效果,实验中大部分数据集的结果均符合这一规律,也就验证了我们算法的合理性和必要性.然而,当原始属性个数与标记个数的比值过于大时(例如在数据集 genbase 中这一比值达到了 43.93∶1),t 值的变化对实验结果的影响就会变的微乎其微.

接下来,我们固定 t 值大小为 0.7,研究基分类器的类型对算法结果的影响.这里我们选择了 3 种基分类器,分别为:决策树 J48、支持向量机(Support Vector Machine,SVM)和朴素贝叶斯分类器(Naive Bayesian,NB).如表 2 所示,对应到 base-level 和 meta-level 中,我们共设计了 6 种组合方式.在 6 个数据上的测试结果如表 3~表 8 所示.

表 2 基分类器选择方法

name	base-level	meta-level
J48+J48	J48	J48
J48+SVM	J48	SVM
J48+NB	J48	LR
SVM+J48	SVM	J48
SVM+SVM	SVM	SVM
SVM+NB	SVM	LR

表 3 Yeast 数据集不同基分类器实验结果

name	*Hamming Loss*	*Recall*	*Precision*	*F1-measure*	*Accuracy*
J48+J48	0.2703	0.5549	0.5254	0.4044	0.6120
J48+SVM	**0.1969**	0.5846	**0.6163**	**0.5071**	**0.6733**
J48+NB	0.3015	0.6127	0.5393	0.4207	0.6674
SVM+J48	0.2605	0.5722	0.5466	0.4231	0.6136
SVM+SVM	0.1989	0.5774	0.6109	0.5006	0.6698
SVM+NB	0.3006	**0.6169**	0.5422	0.4249	0.6710

表 4 Emotions 数据集不同基分类器实验结果

name	*Hamming Loss*	*Recall*	*Precision*	*F1-measure*	*Accuracy*
J48+J48	0.2591	0.5744	0.5282	0.4373	0.6924
J48+SVM	0.1934	0.6278	0.6171	0.5329	0.7506
J48+NB	0.2484	0.7724	0.6360	0.5334	0.7606
SVM+J48	0.2434	0.6100	0.5692	0.4762	0.7014
SVM+SVM	**0.1870**	0.6415	0.6372	**0.5549**	0.7610
SVM+NB	0.2467	**0.7780**	**0.6399**	0.5373	**0.7761**

表 5 Birds 数据集不同基分类器实验结果

name	*Hamming Loss*	*Recall*	*Precision*	*F1-measure*	*Accuracy*
J48+J48	0.0580	0.5229	0.5306	0.5038	0.3888
J48+SVM	0.0615	0.6190	0.6038	0.5668	0.4353
J48+NB	0.3006	0.3137	0.1826	0.1365	0.3641
SVM+J48	**0.0562**	0.5664	0.5724	0.5448	**0.4488**
SVM+SVM	0.0566	**0.6164**	**0.6056**	**0.5722**	0.4316
SVM+NB	0.2998	0.3137	0.1824	0.1364	0.3644

表 6　CAL500 数据集不同基分类器实验结果

name	*Hamming Loss*	*Recall*	*Precision*	*F1-measure*	*Accuracy*
J48＋J48	0.1719	0.3179	0.3507	0.2165	0.3280
J48＋SVM	0.1849	0.3147	0.3379	0.2086	0.2592
J48＋NB	0.2677	**0.4904**	0.3523	0.2218	0.3260
SVM＋J48	0.1800	0.3329	0.3500	0.2162	0.3035
SVM＋SVM	**0.1381**	0.2440	0.3412	0.2104	0.2982
SVM＋NB	0.2473	0.4819	**0.3655**	**0.2322**	**0.3519**

表 7　Genbase 数据集不同基分类器实验结果

name	*Hamming Loss*	*Recall*	*Precision*	*F1-measure*	*Accuracy*
J48＋J48	0.0011	0.9914	0.9904	0.9862	0.9926
J48＋SVM	**0.0008**	**0.9917**	**0.9926**	**0.9894**	**0.9936**
J48＋NB	0.0249	0.5009	0.5209	0.5009	0.8757
SVM＋J48	0.0011	0.9899	0.9893	0.9854	0.9918
SVM＋SVM	0.0008	0.9904	0.9912	0.9882	0.9923
SVM＋NB	0.0247	0.5054	0.5255	0.5054	0.8843

表 8　Medical 数据集不同基分类器实验结果

name	*Hamming Loss*	*Recall*	*Precision*	*F1-measure*	*Accuracy*
J48＋J48	0.0104	0.7988	0.7742	0.7436	**0.8298**
J48＋SVM	0.0102	0.8074	0.7906	0.7618	0.8025
J48＋NB	0.0250	0.4344	0.4040	0.3678	0.6245
SVM＋J48	0.0101	0.8047	0.7864	0.7571	0.8262
SVM＋SVM	**0.0093**	**0.8253**	**0.8086**	**0.7782**	0.8176
SVM＋NB	0.0221	0.5072	0.4731	0.4395	0.6679

从表中数据可以看出，基分类器的选择对实验结果是有比较明显的影响的. 例如，在 genbase 数据集上，J48＋SVM 的基分类器组合可以取得高达 0.99 的 *recall* 值，而 J48＋NB 的基分类器组合的 *recall* 值只有 0.5 左右. 具体到每个数据集的实验结果我们可以发现，总会有这样一种基分类器组合，其在各项评价指标上均表现优异，例如 yeast 数据集上的 J48＋SVM 组合，CAL500 数据集上的 SVM＋NB 组合，medical 数据集上的 SVM＋SVM 组合. 也就是说，不同 base-level/meta-level 基分类器组合适合不同的数据集，为数据集选择合适的基分类器可以提升算法性能.

如 3.2 节中提到的，我们的算法共有 3 种策略：RFS、RFS-S 以及 RFS-J. 我们也设计了实验来比较这 3 种策略的优劣. 这里我们设置 t 值大小为 0.7，base-level 和 meta-level 均选择 SVM 做基分类器，3 种策略在 5 个数据集上的实验结果如表 9～表 13 所示.

表 9　Yeast 数据集不同策略实验结果

strategy	*Hamming Loss*	*Subset Accuracy*	*Recall*	*Precision*	*F1-measure*	*Accuracy*
RFS	0.1989	**0.1481**	0.5774	0.6109	0.5006	0.6698
RFS-S	**0.1986**	**0.1481**	**0.5776**	**0.6112**	**0.5010**	**0.6701**
RFS-J	0.1990	0.1440	0.5743	0.6093	0.4978	0.6678

表 10　Emotions 数据集不同策略实验结果

strategy	*Hamming Loss*	*Subset Accuracy*	*Recall*	*Precision*	*F1-measure*	*Accuracy*
RFS	**0.1870**	**0.3015**	**0.6415**	**0.6372**	**0.5549**	**0.7610**
RFS-S	0.1889	0.2914	0.6289	0.6255	0.5441	0.7523
RFS-J	0.1968	0.2747	0.5877	0.5887	0.5118	0.7301

表 11　Birds 数据集不同策略实验结果

strategy	*Hamming Loss*	*Subset Accuracy*	*Recall*	*Precision*	*F1-measure*	*Accuracy*
RFS	0.0566	**0.4845**	**0.6164**	0.6055	0.5722	0.4316
RFS-S	0.0536	**0.4845**	0.6153	0.6077	**0.5753**	0.4399
RFS-J	**0.0530**	0.4782	0.6158	**0.6086**	0.5741	**0.4547**

表 12 Genbase 数据集不同策略实验结果

strategy	*Hamming Loss*	*Subset Accuracy*	*Recall*	*Precision*	*F1-measure*	*Accuracy*
RFS	0.0008	0.9773	0.9904	0.9912	0.9882	0.9923
RFS-S	**0.0008**	**0.9789**	**0.9920**	**0.9927**	**0.9897**	**0.9938**
RFS-J	0.0008	0.9773	0.9904	0.9912	0.9882	0.9923

表 13 Medical 数据集不同策略实验结果

strategy	*Hamming Loss*	*Subset Accuracy*	*Recall*	*Precision*	*F1-measure*	*Accuracy*
RFS	**0.0093**	**0.6881**	**0.8253**	**0.8086**	**0.7782**	**0.8176**
RFS-S	0.0100	0.6728	0.8077	0.7930	0.7622	0.8029
RFS-J	0.0098	0.6759	0.8091	0.7959	0.7652	0.8049

从实验结果中我们看到，在 emotions 和 medical 数据集上 RFS 策略在各项指标上均领先于其余两种策略，而在 genbase 和 yeast 两个生物领域的数据集上，RFS-S 策略的各项指标全面领先. 这说明，在特定的场景中，分别对原始属性和标记属性进行属性选择是能够提升分类性能的. 然而我们也发现，第三种策略 RFS-J 在各个数据集上的表现普遍不够理想，这可能是由于原始属性和标记属性的语义不同，直接将二者置于同一标准下进行衡量并筛选的策略可能会导致属性选择上的错误. 如何进一步研究原始属性与标记属性间的关系，进而改进 RFS-J 策略有待于我们在今后的工作中继续探索.

第二部分实验中，我们将我们的算法与当下流行的多标记分类算法进行横向比较. 实验中用于比较的基准算法包括传统的 BR 算法，链式算法中的代表算法分类器链 CC[11]，stacking 类算法 DBR[17]，基于关系剪枝的 stacking 算法 CPS[15]. 我们将 CPS 算法中的标记百分比参数 t 设为 0.7，其余所有算法的所有参数都使用 Mulan 平台下的默认参数. 在这部分实验中，我们选择的是策略(1)RFS 算法，属性选择百分比参数 t 同样设为 0.7，base-level 和 meta-level 均选择 SVM 做基分类器. 最终，这 5 种算法的 6 项评价指标在 5 个数据集上的比较结果如表 14～表 19 所示.

表 14 不同算法汉明损失比较结果

dataset	BR	DBR	CPS	CC	RFS
birds	0.0561	0.0577	**0.0538**	0.0562	0.0566
emotions	0.2474	0.2653	0.2589	0.2550	**0.1870**
enron	**0.0508**	0.0519	0.0572	0.0525	**0.0508**
medical	0.0104	0.0104	0.0251	0.0102	**0.0093**
yeast	0.2454	0.2702	0.2237	0.2682	**0.1989**

表 15 不同算法子集正确率比较结果

dataset	BR	DBR	CPS	CC	RFS
birds	0.4469	0.4377	0.4656	0.4501	**0.4845**
emotions	0.1838	0.1466	0.1398	0.2478	**0.3015**
enron	0.1028	0.0922	0.0118	**0.1269**	0.1058
medical	0.6553	0.6522	0.1012	0.6778	**0.6881**
yeast	0.0683	0.0430	0.0600	**0.1531**	0.1481

表 16 不同算法召回率比较结果

dataset	BR	DBR	CPS	CC	RFS
birds	0.5529	0.5192	0.4980	0.5447	**0.6164**
emotions	0.5994	0.5781	0.3838	0.5777	**0.6415**
enron	0.5035	0.4901	0.1856	**0.5091**	0.5042
medical	0.8029	0.7988	0.1270	0.8051	**0.8253**
yeast	**0.5783**	0.5527	0.4738	0.5491	0.5774

表 17 不同算法精确率比较结果

dataset	BR	DBR	CPS	CC	RFS
birds	0.5979	0.5287	0.5314	0.5857	**0.6055**
emotions	0.5808	0.5252	0.4363	0.5738	**0.6372**
enron	**0.6180**	0.5121	0.3445	0.6085	0.5254
medical	0.7792	0.7742	0.1358	0.7892	**0.8086**
yeast	0.6114	0.5279	**0.6955**	0.5633	0.6109

表 18 不同算法 *F1* 测量比较结果

dataset	BR	DBR	CPS	CC	RFS
birds	0.5611	0.5027	0.5061	0.5510	**0.5722**
emotions	**0.5566**	0.4291	0.3703	0.5482	0.5549
enron	0.5257	0.3998	0.2253	**0.5299**	0.4132
medical	0.7771	0.7436	0.1281	**0.7850**	0.7782
yeast	**0.5635**	0.4017	0.5367	0.5279	0.5006

表 19 不同算法准确率比较结果

dataset	BR	DBR	CPS	CC	RFS
birds	**0.5295**	0.3892	0.4933	0.5222	0.4316
emotions	0.4623	0.6860	0.3244	0.4703	**0.7610**
enron	0.4129	0.5877	0.1637	0.4233	**0.5924**
medical	0.7465	**0.8308**	0.1213	0.7581	0.8176
yeast	0.4395	0.6107	0.4161	0.4280	**0.6698**

从表中各个分类评价指标中我们可以看出，我们的 RFS 算法在总体上优于其它 4 种算法. 特别是在汉明损失、子集正确率、准确率这 3 项指标上，RFS 优势较为明显. RF＞BR 说明将标记加入到原始属性空间中可以提升算法精度；RF＞DBR 说明选择部分标记加入到原始属性空间中就可以取得提升算法精度的效果；而 RF＞CPS，则说明我们的算法更能够准确地选择出标记的依赖标记子集加入到原始属性空间中.

5　总结与展望

本文我们从不同的问题转化方法的角度，对标记间依赖关系的利用方法进行了区分. 我们认为，标记间依赖关系的利用方法可以分为标记分组法和属性空间扩展法两种. 我们发现，对于属性空间扩展法，普遍存在的难题在于如何对标记间的依赖关系进行准确度量，并选择合适的标记集合加入到属性空间中. 在此基础上，我们提出了一种基于 ReliefF 剪枝的多标记分类算法. 算法从属性选择的角度，利用 ReliefF 方法对标记间的依赖关系进行度量，进而选择依赖关系较强的标记加入到原始属性空间中. 在实验中，我们对 RFS 算法的内部参数进行了研究，并且将我们的算法与当下先进的多标记分类算法进行了比较. 实验结果验证了我们的算法的有效性.

在本文的工作中，我们也总结得到了一些值得进一步去研究的问题. 例如如何通过建立原始属性与标记属性间的关系，解决二者语义不同的问题，使其能够共同进行属性选择；以及在算法中考虑属性选择代价的问题，优化属性选择步骤等. 这些问题有待于我们在今后的工作中继续探索.

参　考　文　献

[1] Tsoumakas G, Katakis I, Vlahavas I. Mining multi-label data. Data Mining and Knowledge Discovery Handbook. New York, USA: Springer, 2009: 667-685

[2] Katakis I, Tsoumakas G, Vlahavas I. Multilabel text classification for automated tag suggestion//Proceedings of the ECML/PKDD 2008 Discovery Challenge. Antwerp, Belgium, 2008: 75-83

[3] Turnbull D, Barrington L, Torres D, et al. Semantic annotation and retrieval of music and sound effects. IEEE Transactions on Audio, Speech, and Language Processing, 2008, 16(2): 467-476

[4] Snoek C G M, Worring M, Van Gemert J C, et al. The challenge problem for automated detection of 101 semantic concepts in multimedia//Proceedings of the 14th Annual ACM International conference on Multimedia. Santa Barbara, USA, 2006: 421-430

[5] Yang S, Kim S K, Ro Y M. Semantic home photo categorization. IEEE Transactions on Circuits and Systems for Video Technology, 2007, 17(3): 324-335

[6] Zhang M L, Zhou Z H. A review on multi-label learning algorithms. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(8): 1819-1837

[7] Dembczyński K, Waegeman W, Cheng W, et al. On label dependence and loss minimization in multi-label classification. Machine Learning, 2012, 88(1-2): 5-45

[8] Tsoumakas G, Katakis I, Vlahavas I. Random k-labelsets for multilabel classification. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(7): 1079-1089

[9] Rokach L, Schclar A, Itach E. Ensemble methods for multi-label classification. Expert Systems with Applications, 2014, 41(16): 7507-7523

[10] Read J, Pfahringer B, Holmes G. Multi-label classification using ensembles of pruned sets//Proceedings of the 8th IEEE International Conference on Data Mining, 2008(ICDM'08). Pisa, Italy, 2008: 995-1000

[11] Read J, Pfahringer B, Holmes G, et al. Classifier chains for multi-label classification. Machine Learning, 2011, 85(3): 333-359

[12] Cheng W, Hüllermeier E, Dembczynski K J. Bayes optimal multilabel classification via probabilistic classifier chains//Proceedings of the 27th International Conference on Machine Learning(ICML10). Haifa, Israel, 2010: 279-286

[13] Senge R, Coz Velasco J J, Hüllermeier E. Rectifying classifier chains for multi-label classification//Proceedings of the Lernen, Wissen, Adaption(LWA2013). Bamberg, Germany, 2013: 162-169

[14] Godbole S, Sarawagi S. Discriminative methods for multi-labeled classification//Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. Sydney, Australia, 2004: 22-30

[15] Tsoumakas G, Dimou A, Spyromitros E, et al. Correlation-based pruning of stacked binary relevance models for multi-label learning//Proceedings of the 1st International Workshop on Learning from Multi-label Data. Bled, Slovenia, 2009: 101-116

[16] Alvares-Cherman E, Metz J, Monard M C. Incorporating label dependency into the binary relevance framework for multi-label classification. Expert Systems with Applications, 2012, 39(2): 1647-1655

[17] Montañes E, Senge R, Barranquero J, et al. Dependent binary relevance models for multi-label classification. Pattern Recognition, 2014, 47(3): 1494-1508

[18] Spolaôr N, Cherman E A, Monard M C, et al. A comparison of multi-label feature selection methods using the problem transformation approach. Electronic Notes in Theoretical Computer Science, 2013, 292: 135-151

[19] Kononenko I. Estimating attributes: Analysis and extensions of RELIEF//Proceedings of the European Conference on Machine Learning. Catania, Italy, 1994: 171-182

[20] Spolaôr N, Monard M C. Evaluating ReliefF-based multi-label feature selection algorithm//Proceedings of the Ibero-American Conference on Artificial Intelligence. Santiago, Chile, 2014: 194-205

[21] Reyes O, Morell C, Ventura S. Scalable extensions of the ReliefF algorithm for weighting and selecting features on the multi-label learning context. Neurocomputing, 2015, 161: 168-182

[22] Wang Z, Zhang Y, Chen Z, et al. Application of ReliefF algorithm to selecting feature sets for classification of high resolution remote sensing image//Proceedings of the Geoscience and Remote Sensing Symposium (IGARSS). Beijing, China, 2016: 755-758

LIU Hai-Yang, born in 1987, Ph. D. candidate. His research interests include data mining and pattern recognition.

WANG Zhi-Hai, born in 1963, Ph. D., professor, Ph. D. supervisor. His research interests include data mining and machine learning.

ZHANG Zhi-Dong, born in 1992, M. S. candidate. His research interests include data mining and natural language processing.

Background

Multi-label classification (MLC) is a machine learning problem in which models are sought that assign a subset of labels to each instance, unlike conventional (single-class) classification that involves predicting only a single class. The multi-label problem is receiving increased attention and is relevant to many domains such as text categorization, classification of music and videos, semantic annotation of images and many more.

Recently, many studies are looking for efficient and accurate algorithms to cope with multi-label classification challenge. They are usually partitioned into two main categories: algorithm adaptation and problem transformation. In multi-label classification problem the labels will not occur independent of each other; instead, there are statistical dependencies between them. From a learning and prediction point of view, effective exploitation of the label dependencies information is crucial for the success of multi-label learning techniques. These dependencies constitute a promising source of information, in addition to that coming from the mere description of the objects. Indeed, there is an increasing number of papers accounting for possible dependencies between class labels that achieves optimal predictive performance.

In this paper, we divide the utilizing methods of label dependences into two groups from the perspective of different ways of problem transformation: label grouping model and feature space extending model. We find that the common difficulty for both kinds of methods is how to accurately measure the dependences between labels. In particular, for feature space extending methods, how to choose proper labels to extend the original feature space is the key to improve classification performance. On the basis of these, we describe the ReliefF based pruning algorithm for multi-label classification.

The key contribution of this paper is threefold: (1) We provide a new method to measure the dependences between labels. Unlike existing methods measuring pair-wise label dependences, our method related to the ReliefF algorithm takes into account the effect of all interacting labels. (2) Instead of extending the original feature space with all labels, we choose the closely related labels. Thus, we can reduce noise in the data and avoid adverse effects caused by irrelevant labels. (3) In the feature selection phase, we design a brand new strategy that treats original features and label features as the same features and select together.

This work is supported by the National Natural Science Foundation of China (No. 61672086, 61702030, 61771058) and Beijing Natural Science Foundation (No. 4182052).