软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(9):1992−2001 [doi: 10.13328/j.cnki.jos.004641]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于类属属性的多标记学习算法*

吴 磊[1,2], 张敏灵[1,2]

[1](东南大学 计算机科学与工程学院,江苏 南京 210096)

[2](计算机网络和信息集成教育部重点实验室(东南大学),江苏 南京 210096)

通讯作者: 张敏灵, E-mail: zhangml@seu.edu.c

摘 要: 在多标记学习框架中,每个对象由一个示例(属性向量)描述,却同时具有多个类别标记.在已有的多标记学习算法中,一种常用的策略是将相同的属性集合应用于所有类别标记的预测中.然而,该策略并不一定是最优选择,原因在于每个标记可能具有其自身独有的特征.基于这个假设,目前已经出现了基于标记的类属属性进行建模的多标记学习算法 LIFT.LIFT 包含两个步骤:属属性构建与分类模型训练.LIFT 首先通过在标记的正类与负类示例上进行聚类分析,构建该标记的类属属性;然后,使用每个标记的类属属性训练对应的二类分类模型.在保留 LIFT 分类模型训练方法的同时,考察了另外 3 种多标记类属属性构造机制,从而实现 LIFT 算法的 3 种变体——LIFT-MDDM,LIFT-INSDIF 以及 LIFT-MLF.在 12 个数据集上进行了两组实验,验证了类属属性对多标记学习系统性能的影响以及 LIFT 采用的类属属性构造方法的有效性.

关键词: 机器学习;多标记学习;类属属性;降维;标记相关性

中图法分类号: TP181

中文引用格式: 吴磊,张敏灵.基于类属属性的多标记学习算法.软件学报,2014,25(9):1992−2001. http://www.jos.org.cn/1000-9825/4641.htm

英文引用格式: Wu L, Zhang ML. Research of label-specific features on multi-label learning algorithm. Ruan Jian Xue Bao/Journal of Software, 2014,25(9):1992−2001 (in Chinese). http://www.jos.org.cn/1000-9825/4641.htm

Label-Specific Features on Multi-Label Learning Algorithm

WU Lei[1,2], ZHANG Min-Ling[1,2]

[1](Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

[2](Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing 210096, China)

Corresponding author: ZHANG Min-Ling, E-mail: zhangml@seu.edu.c

Abstract: In the framework of multi-label learning, each example is represented by a single instance (feature vector) while simultaneously associated with multiple class labels. A common strategy adopted by most existing multi-label learning algorithms is that the very feature set of each example is employed in the discrimination processes of all class labels. However, this popular strategy might be suboptimal as each label is supposed to possess specific characteristics of its own. Based on this assumption, a multi-label learning algorithm named LIFT is proposed, in which label specific feature of each label is utilized in the discrimination process of the corresponding label. LIFT contains two steps: label-specific features construction and classification models induction. LIFT constructs the label-specific features by querying the clustering results and then induces the classification model with the corresponding label-specific features. In this paper, three variants of LIFT are studied, all employ other label-specific feature construction mechanisms while retaining the classification models induction process of LIFT. To validate the general helpfulness of label-specific feature mechanism to multi-label learning and the effectiveness of those label-specific features adopted by LIFT, two groups of experiments are conducted on a total of twelve multi-label benchmark datasets.

* 基金项目: 国家自然科学基金(61175049, 61222309); 教育部新世纪优秀人才支持计划(NCET-13-0130)
收稿时间: 2014-04-04; 定稿时间: 2014-05-14

© 中国科学院软件研究所 http://www.jos.org.cn

Key words: machine learning; multi-label learning; label specific feature; dimensionality reduction; label correlation

在多标记学习框架下,对于现实世界中的一个对象,学习系统在输入空间用一个示例(属性向量)刻画该对象的性质,在输出空间使用类别标记表征该对象的语义信息.多标记学习的目标是为现实世界中同时具有多个语义信息的对象建立分类模型,使得该模型可以有效地预测未见示例的所有相关类别标记[1].在实际应用中,存在很多同时具有多种语义信息的对象,例如:在文档分类应用中,一篇具有标记“科技”的文档,很有可能同时具有标记“经济”[2−4];多媒体内容标注应用中,一幅图像可能同时具有标记“大海”、“天空”以及“帆船”等[5−7].

形式化地,假设 $X=\mathbb{R}^d$ 代表 d 维的示例空间,$y=\{l_1,l_2,\ldots,l_q\}$代表由 q 个类别标记组成的标记空间,多标记学习系统的目标就是利用多标记训练集 $D=\{(\boldsymbol{x}_i,Y_i)|1\leqslant i\leqslant m\}$学习得到一个分类器 $h:X\rightarrow 2^y$.其中,对于每一个多标记样本$(\boldsymbol{x}_i,Y_i)$,$\boldsymbol{x}_i\in X$ 为 d 维特征向量$(x_{i1},x_{i2},\ldots,x_{id})^T$,$Y_i\subseteq y$ 为示例 $\boldsymbol{x}_i$ 的相关标记集合.对于任何一个未见示例 $\boldsymbol{x}\in X$,多标记分类器 $h(\cdot)$可以预测其相关标记集合 $h(\boldsymbol{x})\subseteq y$.在很多情况下,多标记学习系统的输出对应于一组实值函数$\{f_1,f_2,\ldots,f_q\}$,其中,函数 $f_k:X\rightarrow\mathbb{R}(1\leqslant k\leqslant q)$的返回值用于确定标记 l_k 是否为当前待测示例的相关标记.此时,上述的多标记分类器对应于 $h(\boldsymbol{x})=\{l_k|f_k(\boldsymbol{x})>0,1\leqslant k\leqslant q\}$.在过去的 10 年中,涌现出了大量的多标记学习算法[8],这些算法采用的共同策略是使用相同的属性集合预测所有的类别标记.换句话说,多标记学习系统返回的 q 个实值函数$\{f_1,f_2,\ldots,f_q\}$是由相同的属性集合训练得到的.

尽管上述策略在算法设计上取得了很大的成功,但是它可能并非最佳选择.举例来说,在图像语义标注系统中,直观上,我们可以认为颜色相关的属性(color-based feature)在判断“蓝天”这一标记上会有好的效果,而纹理相关的属性(texture-based feature)在判断“沙漠”这一标记上会有好的效果;在文本分类应用中,包含“政府”、“大选”等词汇的属性在判断“政治”这一标记上会有好的效果,而包含“GDP”、“纳税”等词汇的属性在判断“经济”这一标记上会有好的效果.上面的例子反映了一个事实,即,一个对象具有多种语义信息的原因在于该对象包含了描述这些语义信息的属性.换言之,每个标记可能具有其自身独特的属性特征,这些属性也是与该标记最相关、对该标记最具有判别能力的属性,称为类属属性(label-specific feature).基于此,文献[9]中提出了一种基于类属属性的多标记学习算法 LIFT.该算法分为两个步骤:类属属性构造与分类模型训练.首先,LIFT 在每个标记 $l_k\in y$ 的正类训练样本和负类训练样本上分别进行聚类分析,利用聚类结果构建标记 l_k 的类属属性;然后,利用每个标记的类属属性为该标记训练一个二类分类模型 $f_k:X\rightarrow\mathbb{R}(1\leqslant k\leqslant q)$,以此完成多标记学习任务.

本文在保留 LIFT 分类模型训练方法的同时,考察了另外 3 种多标记类属属性构造机制,从而实现 LIFT 的 3 种变体——LIFT-INSDIF,LIFT-MDDM 以及 LIFT-MLF.通过在 12 个数据集上进行的两组实验,验证了类属属性对多标记学习系统性能的影响以及 LIFT 采用的类属属性构造方法的有效性.

本文第 1 节介绍文献[9]中提出的 LIFT 算法.第 2 节给出多标记学习中已有的 3 种属性转换方法,并基于此设计 LIFT 的 3 种变体算法.第 3 节是本文的实验部分,包括算法参数设置、数据集、结果分析等.最后一节对本文工作进行总结.

1 LIFT 算法

多标记学习的最大挑战在于其输出空间的类别标记集合数将随着标记个数的增加成指数规模增长.因此,为有效应对输出空间过大带来的挑战,如何利用标记之间的相关性提高系统的泛化性能,被认为是多标记学习取得成功的关键[8,10].基于此,已有多标记学习算法的一个共同点在于:重点关注输出空间(标记空间)[11,12],即考察标记之间的相关性,而较少关注输入空间(属性空间).与已有策略不同,文献[9]通过构建类属属性对输入空间的性质进行了考察,提出了多标记学习算法 LIFT.

给定多标记训练集 $D=\{(\boldsymbol{x}_i,Y_i)|1\leqslant i\leqslant m\}$,其中,对于每一个多标记样本$(\boldsymbol{x}_i,Y_i)$,$\boldsymbol{x}_i\in X$ 为 d 维属性向量$(\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\ldots,\boldsymbol{x}_{id})^T$,$Y_i\subseteq y$ 是示例 $\boldsymbol{x}_i$ 的相关标记集合.基于此,LIFT 算法的的两个步骤可按如下方式完成.

1) 类属属性构建

在该步骤中,LIFT 的目的在于构建每个标记的类属属性,以提高多标记学习系统的学习性能.为了达到这

© 中国科学院软件研究所 http://www.jos.org.cn

个目的,需要考察每个标记所对应样本的内在性质.具体来说,对于标记 $l_k \in y$,其正类示例集合与负类示例集合分别为

$$P_k = \{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, Y_i) \in D, l_k \in Y_i\}$$
$$N_k = \{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, Y_i) \in D, l_k \notin Y_i\} \tag{1}$$

换句话说,P_k 与 N_k 分别是由具有标记 l_k 的示例与不具有标记 l_k 的示例组成的集合.

为了考察集合 P_k 与 N_k 的内在性质,LIFT 在两个集合上分别进行聚类分析(clustering analysis).文献[9]中采用简单且高效的 k-means 算法进行聚类分析.在此,将集合 P_k 的 m_k^+ 个聚类中心记为 $\{\boldsymbol{p}_1^k, \boldsymbol{p}_2^k, ..., \boldsymbol{p}_{m_k^+}^k\}$.相应地,将集合 N_k 的 m_k^- 个聚类中心记为 $\{\boldsymbol{n}_1^k, \boldsymbol{n}_2^k, ..., \boldsymbol{n}_{m_k^-}^k\}$.为了应对可能存在的类别不平衡问题(即$|P_k|<<|N_k|$)[8],LIFT 将 P_k 与 N_k 的聚类中心个数设置为相等,即 $m_k^+ = m_k^- = m_k$.如此,可以认为是对正类示例与负类示例的聚类信息赋予相同的权重.具体来说,LIFT 将示例集合 P_k 与 N_k 上的聚类个数设定为

$$m_k = \lceil r \cdot \min(|P_k|, |N_k|) \rceil \tag{2}$$

其中,$|\cdot|$返回集合的势,且 $r \in [0,1]$是控制聚类个数的参数.

由聚类的性质可知,P_k 与 N_k 的聚类中心刻画了这两个集合中数据的内在结构.因此,聚类中心可以被用于构建标记的类属属性.本质上,类属属性的构建对应于一个从原 d 维属性空间 X 到新的 d'维属性空间 Z_k 的映射 $\varphi_k: X \to Z_k (1 \leqslant k \leqslant q)$.LIFT 按如下公式定义映射 φ_k:

$$\varphi_k(\boldsymbol{x}) = [d(\boldsymbol{x}, \boldsymbol{p}_1^k), ..., d(\boldsymbol{x}, \boldsymbol{p}_{m_k}^k), d(\boldsymbol{x}, \boldsymbol{n}_1^k), ..., d(\boldsymbol{x}, \boldsymbol{n}_{m_k}^k)] \tag{3}$$

其中,$d(\cdot,\cdot)$返回两向量之间的距离,文献[9]中设置为欧氏距离.

2) 分类模型训练

在该步骤中,LIFT 使用上一步中构造的类属属性分别训练 q 个分类模型$\{f_1, f_2, \ldots, f_q\}$.参照形式化定义,对于标记 $l_k \in y$,首先根据映射 φ_k 从原训练集 D 构造其相应的二类训练集 BT_k:

$$BT_k = \{(\varphi_k(\boldsymbol{x}_i), \phi(Y_i, l_k)) \mid (\boldsymbol{x}_i, Y_i) \in D\} \tag{4}$$

其中,若 $l_k \in Y_i$,则 $\phi(Y_i, l_k)$返回+1;否则,返回−1.基于二类训练集 BT_k,可以使用任何一种二类分类算法 B 学习得到标记 l_k 对应的分类模型 $f_k: Z_k \to \mathbb{R}$.

对于未见示例 $u \in X$,其相关标记按如下方式预测:

$$Y = \{l_k \mid f_k(\varphi_k(u)) > 0, 1 \leqslant k \leqslant q\} \tag{5}$$

LIFT 的伪码见文献[9].

2 多标记类属属性机制

直观上看,解决多标记学习问题最简单的方法是:将多标记学习问题转换成为 q 个相互独立的二类分类问题,每一个二类分类问题对应于标记空间中的一个类别标记[5].基于此,Boutell 等人在文献[5]中提出了多标记学习算法 Binary Relevance(BR).针对第 k 个标记 l_k,BR 首先构建该标记的二类训练集 $D_k = \{(\boldsymbol{x}_i, \phi(Y_i, l_k)) \mid 1 \leqslant i \leqslant m\}$.基于此,使用某种已有的二类分类算法 B 训练一个二类分类器 $g_k: X \to \mathbb{R}$.在预测阶段,对于未见示例 $\boldsymbol{x}$,BR 将 q 个二类分类器上的预测结果作为该未见示例的相关标记集合,即,$Y = \{l_k \mid g_k(\boldsymbol{x}) > 0, 1 \leqslant k \leqslant q\}$.

由此可见,BR 中各二类分类模型的训练均基于相同的属性向量,即针对每个标记的类属属性映射为 $\varphi_k: X \to X (1 \leqslant k \leqslant q)$.因此,BR 可以看作 LIFT 算法的一个退化版本(degenerated version).

如第 1 节所述,多标记类属属性机制本质上对应于构建从原 d 维属性空间 X 到新的 d'维属性空间 Z_k 的映射 $\varphi_k: X \to Z_k (1 \leqslant k \leqslant q)$.下面,本文将给出 3 种基于多标记类属属性机制的 LIFT 变体算法——LIFT-INSDIF, LIFT-MDDM 以及 LIFT-MLF.其中,对应于每个标记的二类分类模型使用不同的属性向量进行训练,而其二类分类模型的训练过程与 LIFT 保持一致.

- LIFT-MDDM

多标记降维是一种常用的多标记属性转换方式.在此,我们使用已提出的一种基于过滤策略的多标记降维

© 中国科学院软件研究所 http://www.jos.org.cn

算法 MDDM[13]构建 LIFT 的变体算法 LIFT-MDDM.MDDM 使用希尔伯特-施密特独立标准(Hilbert- schmidt independence criterion,简称 HSIC)度量原始属性空间 X 和标记空间 y 的关联度(dependency),通过最大化该关联度求得 $d\times d'$维线性转换矩阵 $\boldsymbol{P}(d'<<d)$.最终,LIFT-MDDM 针对每个标记的类属属性映射可通过矩阵 $\boldsymbol{P}$ 构建,即

$$\varphi_k(\boldsymbol{x})=\boldsymbol{P}^T\boldsymbol{x}(1\leqslant k\leqslant q) \tag{6}$$

- LIFT-INSDIF

如本文引言所述,一个对象具有多种语义信息的原因在于,该对象同时包含了描述这些语义的属性.因此,相比于使用单示例描述方式,使用多个示例描述一个对象能够更好地刻画该对象的多重语义信息[14].基于此,我们使用多标记学习算法 INSDIF[15]中提出的属性转换方法构建 LIFT 的变体算法 LIFT-INSDIF.通过将示例 $\boldsymbol{x}$ 与每个标记的正类样本集 $P_k=\{\boldsymbol{x}_i|(\boldsymbol{x}_i,Y_i)\in D,l_k\in Y_i\}$的中心求差,INSDIF 将示例 $\boldsymbol{x}$ 转换成为由 q 个示例组成的示例包,即,$\{\boldsymbol{x}^{(k)}|1\leqslant k\leqslant q\}$,其中, $\boldsymbol{x}^{(k)}=\boldsymbol{x}-\frac{1}{|P_k|}\sum_{p\in P_k}\boldsymbol{p}$.由此可得 LIFT-INSDIF 中针对每个标记的类属属性映射:

$$\varphi_k(\boldsymbol{x})=\boldsymbol{x}^{(k)}=\boldsymbol{x}-\frac{1}{|P_k|}\sum_{p\in P_k}\boldsymbol{p} \tag{7}$$

- LIFT-MLF

另一种多标记属性转换机制是由原属性空间 X 构建元级属性(meta-level feature),以此刻画示例和标记之间的关系.在此,我们使用算法 MLF[16]中的元级属性构建方法,实现 LIFT 的变体算法 LIFT-MLF.MLF 将每个示例 $\boldsymbol{x}$ 转换为 $q\cdot(3r+2)$维的元级属性向量$[\psi(\boldsymbol{x},l_1),\ldots,\psi(\boldsymbol{x},l_q)]$,其中$\psi(\boldsymbol{x},l_k)(1\leqslant k\leqslant q)$是由示例 $\boldsymbol{x}$ 与 P_k 中的 r 个近邻样本构造的 $3r+2$ 维元级属性.具体来说,$\psi(\boldsymbol{x},l_k)$是示例 $\boldsymbol{x}$ 与其 r 个近邻样本的 L2 距离、L1 距离、余弦距离以及与 P_k 中心的 L2 距离、余弦距离的结合.由此可得 LIFT-MLF 中针对每个标记的类属属性映射:

$$\varphi_k(\boldsymbol{x})=\psi(\boldsymbol{x},l_k) \tag{8}$$

由此可见,以上介绍的 BR,LIFT-MDDM,LIFT-INSDIF,LIFT-MLF 以及 LIFT 都是一阶算法[8],未考虑标记之间的相关性.同时,上述算法的区别在于类属属性的映射方式不同,其二类分类模型的训练过程完全一致.

3 实 验

3.1 数据集

对于数据集 $S=\{(\boldsymbol{x}_i,Y_i)|1\leqslant i\leqslant p\}$,我们使用$|S|$,$dim(S)$,$L(S)$及 $F(S)$分别表示样本个数、属性个数、标记个数以及属性类型.除此之外,在此还给出了几种描述多标记数据集性质的统计量[8,17]:

- $LCard(S)=\frac{1}{p}\sum_{i=1}^{p}|Y_i|$:标记基数(label cardinality),即每个样本具有的平均相关标记个数;
- $LDen(S)=\frac{1}{L(S)}\cdot LCard(S)$:标记密度(label density),即由标记个数归一化的标记基数;
- $LDiv(S)=|\{Y|(\boldsymbol{x},Y)\in S\}|$:标记多样性(label diversity),即数据集中不同标记集合的个数;
- $PLDiv(S)=\frac{1}{|S|}\cdot LDiv(S)$:归一化标记多样性,即由样本个数归一化的相异标记集.

表 1 按照样本数的多少概括了本文实验所用的数据集,其中包括 6 个常规规模的数据集($|S|\leqslant 5000$)以及 6 个大规模的数据集($|S|>5000$).此外,根据惯例对文本数据集 rcv1(subset1),rcv1(subset2),eurlex(subject matter)以及 tmc2007 按照文档频率[18]进行了降维处理.其中,文档频率最高的 2%的属性被保留作为最终属性.

从表 1 可以看出:这 12 个数据集涵盖了很多实际应用领域,且其多标记性质各不相同.由此可见,本文实验所用的数据集是比较充分且全面的,具有较强的概括性.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 1 Characteristics of experimental data sets

表 1 实验数据集性质

Data set	$\|S\|$	$dim(S)$	$L(S)$	$F(S)$	$LCard(S)$	$LDen(S)$	$LDiv(S)$	$PLDiv(S)$	Domain	URL *
enron	1 702	1 001	53	nominal	3.378	0.064	753	0.442	text	URL 2
image	2 000	294	5	numeric	1.236	0.247	20	0.010	image	URL 3
scene	2 407	294	6	numeric	1.074	0.179	15	0.006	image	URL 1
yeast	2 417	103	14	numeric	4.237	0.303	198	0.082	biology	URL 3
slashdot	3 782	1 079	22	nominal	1.180	0.054	156	0.041	text	URL 2
corel5k	5 000	499	374	nominal	3.522	0.009	3 175	0.635	image	URL 1
rcv1 (subset1)	6 000	944	101	numeric	2.880	0.029	1 028	0.171	text	URL 1
rcv1 (subset2)	6 000	944	101	numeric	2.634	0.026	954	0.159	text	URL 2
bibtex	7 395	1 836	159	nominal	2.402	0.015	2 856	0.386	text	URL 3
corel16k (sample 1)	13 766	500	153	nominal	2.859	0.019	4 803	0.349	image	URL 4
eurlex (subject matter)	19 348	100	201	numeric	2.213	0.011	2 504	0.129	text	URL 5
tmc2007	28 596	981	22	nominal	2.158	0.098	1 341	0.047	text	URL 6

URL 1: http://mulan.sourceforge.net/datasets.html
URL 2: http://meka.sourceforge.net/#datasets
URL 3: http://cse.seu.edu.cn/PersonalPage/zhangml/index.htm

3.2 实验设置

由于每个样本可能具有多个标记,故传统的单标记评价指标如精度(accuracy)、查准率(precision)、查全率(recall)等在多标记学习中不再适用[8].假设多标记测试集为 $T=\{(\boldsymbol{x}_i,Y_i)|1\leqslant i\leqslant t\}$,多标记学习系统返回的 q 个实值函数为 $\{f_1,f_2,\ldots,f_q\}$.在本文的实验中,我们使用 5 种基于样本的多标记评价指标[3,19,20]来评价学习系统的性能:

- Hamming Loss

$$HLoss_T(h)=\frac{1}{t}\sum_{i=1}^{t}|h(\boldsymbol{x}_i)\Delta Y_i|,$$

其中,$h(\boldsymbol{x}_i)=\{l_k|f_k(\boldsymbol{x}_i)>0,1\leqslant k\leqslant q\}$ 对应于示例 $\boldsymbol{x}_i$ 的相关标记集合,Δ 返回两集合之间的对称差(symmetric difference).该评价指标考察样本在单个标记上的误分类情况,即:相关标记被预测为无关标记,或者无关标记被预测为相关标记.该评价指标取值越小,则系统性能越优,其最优值为 0.

- Ranking Loss

$$RLoss_T(f)=\frac{1}{t}\sum_{i=1}^{t}\frac{1}{|Y_i||\overline{Y}_i|}|\{(l_a,l_b)|f_a(\boldsymbol{x}_i)\leqslant f_b(\boldsymbol{x}_i),(l_a,l_b)\in Y_i\times\overline{Y}_i\}|.$$

该评价指标衡量错误排序的标记对的比例,即,无关标记排在相关标记之前的比例.该评价指标取值越小,则系统性能越优,其最优值为 0.

- One-Error

$$One\text{-}Error_T(f)=\frac{1}{t}\sum_{i=1}^{t}[\![[\arg\max_{l\in y}f_l(\boldsymbol{x}_i)]\notin Y_i]\!],$$

其中,当谓词 π 成立时,$[\![\pi]\!]$ 返回 1;否则返回 0.该评价指标衡量在样本的类别标记排序中,排在第 1 位的标记不是相关标记的样本所占的比例.该评价指标越小,则系统性能越优,其最优值为 0.

- Coverage

$$Coverage_T(f)=\frac{1}{t}\sum_{i=1}^{t}\max_{l\in Y_i}rank_f(\boldsymbol{x}_i,l)-1,$$

其中,$rank_f(\boldsymbol{x}_i,l)$ 对应于标记 l 根据 $\{f_1(\boldsymbol{x}_i),f_2(\boldsymbol{x}_i),\ldots,f_q(\boldsymbol{x}_i)\}$ 的值降序排序后所处的位置.该评价指标衡量在测试集 T 上遍历到样本所有相关标记的平均查找深度.该评价指标取值越小,则系统性能越优,其最优值为 1.

- Average Precision

$$AvgPrec_T(f)=\frac{1}{t}\sum_{i=1}^{t}\frac{1}{|Y_i|}\sum_{l\in Y_i}\frac{|\{l'|rank_f(\boldsymbol{x}_i,l')\leqslant rank_f(\boldsymbol{x}_i,l),l'\in Y_i\}|}{rank_f(\boldsymbol{x}_i,l)}.$$

该评价指标考察在样本的标记排序序列中,排在该样本相关标记之前的标记仍为相关标记的比例.该评价

© 中国科学院软件研究所 http://www.jos.org.cn

指标取值越大,则系统性能优越,其最优值为 1.

为便于对比,本文通过除以标记个数 $L(S)$将评价指标 Coverage 规格化到区间[0,1].上述基于样本的评价指标首先衡量学习系统在每个测试样本上的分类性能,然后返回整个测试集的均值作为最终结果.

此外,本文还采用一种基于标记的评价指标 Macro-averaging AUC[21].与基于样本的评价指标不同,基于标记的评价指标首先衡量学习系统在每个标记上分类性能,然后返回在所有标记上的均值作为最终结果.

- Macro-averaging AUC

$$AUC_{\text{macro}}=\frac{1}{q}\sum_{k=1}^{q}\frac{|\{(\boldsymbol{x}',\boldsymbol{x}'')\mid f_k(\boldsymbol{x}')\geqslant f_k(\boldsymbol{x}''),(\boldsymbol{x}',\boldsymbol{x}'')\in T_k\times\overline{T}_k\}|}{|T_k||\overline{T}_k|},$$

其中,$T_k=\{\boldsymbol{x}_i|l_k\in Y_i,1\leqslant i\leqslant t\}$,$\overline{T}_k=\{\boldsymbol{x}_i\mid l_k\notin Y_i,1\leqslant i\leqslant t\}$ 分别对应于含有标记 l_k 的测试样本集合与不含有标记 l_k 的测试样本集合.值得注意的是,上述公式可由 AUC 与 Wilcoxon-Mann-Whitney 统计量的相互关系导出[22].此外,该评价指标取值越大,系统的性能越优,其最优值为 1.

本文将给出 6 种算法在上述 12 个数据集上的实验结果,包括 BR,ML-*k*NN,LIFT,LIFT-MDDM,LIFT-INSDIF 以及 LIFT-MLF.根据文献[9],设定 LIFT 的参数 r=0.1;根据文献[23],设定 ML-*k*NN 的近邻个数 k=10;对于 LIFT-MDDM,为与原文保持一致,控制转换后特征空间维度(即 d')的参数 *thr* 设定为 99.9%;对于 LIFT-MLF,我们采取原文的策略,对近邻个数 r 在区间[10,100]进行调节,并最终设定为 10.此外,为了对比的公平,所有算法的基二类分类器均使用线性核 LIBSVM[24].

3.3 结果分析

在每个数据集中,我们采用无放回抽样选取 50%的样本构成训练集,余下的 50%样本构成测试集,抽样过程重复 10 次并记录 10 次实验的均值和方差.

本文共有两组实验:第 1 组将没有使用类属属性机制的 BR 以及常用的 ML-*k*NN 算法与采用类属属性机制的算法 LIFT,LIFT-MDDM,LIFT-INSDIF 以及 LIFT-MLF 进行对比,目的在于验证类属属性机制对多标记学习的影响;第 2 组将 LIFT 与算法 LIFT-MDDM,LIFT-INSDIF,LIFT-MLF 进行对比,目的在于验证 LIFT 所采用的类属属性构建方法的有效性.

表 2 与表 3 给出了 6 个算法在 12 个数据集上的实验结果,实验结果采用均值±方差(mean±std)的形式表示.对于每个评价指标,↑表示其值越大,性能越优;相应地,↓表示其值越小,性能越优.此外,对比算法中性能的最好的结果使用黑体标出.

对于第 1 组实验,我们分别统计了 BR,ML-*k*NN 与 LIFT 及其变体算法在 12 个数据集、6 个评价指标上的 72 个对比结果.与 LIFT 和 LIFT-INSDIF 相比,BR 没有胜出的情况;与 LIFT-MDDM 相比,胜出的情况仅占 5.56%;与 LIFT-MLF 相比,胜出的情况占 44.44%.与 LIFT,LIFT_INSDIF 以及 LIFT_MDDM 相比,ML-*k*NN 胜出的情况分别仅占 6.9%,15.3%以及 30.6%;与 LIFT_MLF 相比,胜出的情况占 66.7%.从 BR,ML-*k*NN 与 LIFT,LIFT-INSDIF,LIFT-MDDM 的对比结果可以看出,有效的类属属性可以较大程度地提高多标记学习算法的学习性能.值得注意的是,LIFT-MLF 的性能在一些数据集上比 BR,ML-*k*NN 的性能差.一个可能的原因是,该算法使用近邻样本构造元级属性的方式不能有效地构造标记的类属属性.可见,并非所有的属性转换机制都能十分有效地地构造类属属性.

如第 2 节所述,BR 可认为是 LIFT 及其变体算法的退化(degenerated)版本.为了进一步验证类属属性对多标记学习算法的影响,对于每个评价指标,本文使用成对 T 检验(检验水平 α=0.05)对 BR 与 LIFT 及其变体算法进行统计分析(见表 4(a)).表 4(a)中的结果进一步证实了上述结论,即:类属属性是提高多标记学习系统泛化性能的可行途径,但并非所有的多标记属性转换机制都能有效地构建类属属性.

与第 1 组实验相同,我们统计了 LIFT 及其 3 个变体算法在 12 个数据集、6 个评价指标上的 72 个对比结果.其中,LIFT 排在第 1 位的情况占 87.5%,排在第 2 位和第 3 位的情况分别仅占 11.11%和 1.39%,且没有排在最后一位与倒数第 2 位的情况发生.由以上统计可以看出,LIFT 的学习性能优秀,进而说明 LIFT 的类属属性构造方法是非常有效的.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 2 Predictive performance of each comparing algorithm (mean±std) on the six regular-scale data sets

表 2 对比算法在 6 个常规规模数据集上的实验结果(mean±std)

Criteria	Algorithms	enron	image	scene	yeast	slashdot	corel5k
Hamming loss ↓	BR	0.060±0.001	0.185±0.004	0.111±0.003	0.201±0.003	0.049±0.001	0.012±0.001
	ML-*k*NN	0.055±0.001	0.186±0.005	0.092±0.003	0.206±0.001	0.053±0.001	**0.009±0.001**
	LIFT-INSDIF	0.053±0.001	0.184±0.002	0.109±0.002	0.201±0.002	0.048±0.001	0.010±0.001
	LIFT	**0.048±0.001**	**0.163±0.003**	**0.084±0.002**	**0.197±0.002**	**0.040±0.001**	0.010±0.001
	LIFT-MDDM	0.049±0.001	0.205±0.004	0.107±0.002	0.206±0.003	0.046±0.001	0.010±0.001
	LIFT-MLF	0.102±0.019	0.245±0.002	0.151±0.005	0.224±0.002	0.120±0.018	0.010±0.001
Ranking loss ↓	BR	0.308±0.007	0.285±0.009	0.171±0.005	0.315±0.005	0.216±0.005	0.655±0.004
	ML-*k*NN	0.100±0.002	0.194±0.008	0.085±0.004	0.182±0.003	0.178±0.005	0.137±0.002
	LIFT-INSDIF	0.091±0.002	0.177±0.007	0.094±0.006	0.174±0.003	0.120±0.003	0.135±0.002
	LIFT	**0.084±0.003**	**0.155±0.007**	**0.069±0.002**	**0.169±0.004**	**0.100±0.003**	**0.131±0.003**
	LIFT-MDDM	0.095±0.003	0.223±0.016	0.089±0.003	0.180±0.004	0.117±0.005	0.142±0.002
	LIFT-MLF	0.350±0.070	0.364±0.067	0.154±0.012	0.197±0.003	0.311±0.031	0.230±0.008
One-Error ↓	BR	0.498±0.012	0.406±0.012	0.348±0.007	0.256±0.008	0.501±0.007	0.849±0.008
	ML-*k*NN	0.328±0.013	0.356±0.010	0.246±0.009	0.247±0.010	0.670±0.017	0.744±0.012
	LIFT-INSDIF	0.289±0.018	0.333±0.013	0.269±0.015	0.234±0.008	0.439±0.006	**0.696±0.009**
	LIFT	**0.254±0.005**	**0.289±0.012**	**0.212±0.011**	**0.229±0.011**	0.430±0.013	0.706±0.012
	LIFT-MDDM	0.262±0.016	0.422±0.026	0.263±0.009	0.236±0.007	**0.406±0.010**	0.697±0.004
	LIFT-MLF	0.627±0.084	0.515±0.077	0.407±0.027	0.252±0.008	0.789±0.075	0.728±0.014
Coverage ↓	BR	0.595±0.010	0.280±0.008	0.158±0.004	0.641±0.005	0.238±0.005	0.898±0.003
	ML-*k*NN	0.265±0.006	0.209±0.008	0.084±0.004	0.465±0.005	0.191±0.005	0.312±0.004
	LIFT-INSDIF	0.249±0.006	0.194±0.006	0.093±0.004	**0.458±0.005**	0.117±0.003	**0.294±0.005**
	LIFT	**0.245±0.007**	**0.178±0.007**	**0.071±0.002**	**0.458±0.007**	**0.115±0.003**	0.313±0.008
	LIFT-MDDM	0.266±0.007	0.232±0.013	0.088±0.002	0.472±0.006	0.117±0.005	0.309±0.004
	LIFT-MLF	0.654±0.064	0.318±0.047	0.142±0.010	0.477±0.004	0.271±0.030	0.496±0.016
Average precision ↑	BR	0.449±0.011	0.709±0.008	0.771±0.005	0.672±0.005	0.572±0.005	0.101±0.003
	ML-*k*NN	0.609±0.010	0.769±0.007	0.853±0.005	0.744±0.005	0.481±0.012	0.240±0.005
	LIFT-INSDIF	0.648±0.007	0.785±0.008	0.838±0.009	0.756±0.005	0.666±0.005	0.267±0.002
	LIFT	**0.685±0.005**	**0.811±0.007**	**0.875±0.006**	**0.763±0.006**	0.671±0.008	**0.280±0.004**
	LIFT-MDDM	0.677±0.007	0.731±0.016	0.844±0.005	0.747±0.007	**0.687±0.006**	0.267±0.002
	LIFT-MLF	0.352±0.067	0.635±0.056	0.750±0.017	0.721±0.006	0.339±0.014	0.248±0.006
Macro-Avg AUC ↑	BR	0.579±0.007	0.705±0.007	0.801±0.003	0.565±0.003	0.656±0.009	0.518±0.001
	ML-*k*NN	0.606±0.007	0.813±0.006	0.927±0.003	0.645±0.007	0.586±0.006	0.530±0.003
	LIFT-INSDIF	0.650±0.016	0.818±0.005	0.910±0.003	0.642±0.014	0.834±0.013	0.600±0.007
	LIFT	**0.688±0.018**	**0.844±0.006**	**0.943±0.002**	**0.673±0.007**	**0.847±0.009**	**0.679±0.006**
	LIFT-MDDM	0.659±0.015	0.782±0.010	0.914±0.004	0.629±0.012	0.826±0.019	0.631±0.007
	LIFT-MLF	0.498±0.002	0.610±0.031	0.828±0.007	0.574±0.007	0.496±0.001	0.586±0.009

Table 3 Predictive performance of each comparing algorithm (mean±std) on the six large-scale data sets

表 3 对比算法在 6 个大规模数据集上的实验结果(mean±std)

Criteria	Algorithms	rcv1 (subset1)	rcv1 (subset2)	bibtex	corel16k (sample1)	eurlex (subject matter)	tmc2007
Hamming loss ↓	BR	0.031±0.001	0.028±0.001	0.015±0.001	0.020±0.001	**0.008±0.001**	0.071±0.001
	ML-*k*NN	0.028±0.001	0.025±0.001	0.014±0.001	**0.019±0.001**	**0.008±0.001**	0.075±0.001
	LIFT-INSDIF	0.027±0.001	0.024±0.001	**0.013±0.001**	**0.019±0.001**	**0.008±0.001**	0.066±0.001
	LIFT	**0.026±0.001**	**0.023±0.001**	**0.013±0.001**	**0.019±0.001**	**0.008±0.001**	**0.061±0.001**
	LIFT-MDDM	0.027±0.001	0.024±0.001	**0.013±0.001**	**0.019±0.001**	0.009±0.001	0.063±0.001
	LIFT-MLF	0.027±0.001	0.024±0.001	0.014±0.001	0.020±0.001	0.033±0.009	0.115±0.018
Ranking loss ↓	BR	0.279±0.004	0.251±0.004	0.303±0.004	0.422±0.001	0.463±0.003	0.216±0.003
	ML-*k*NN	0.089±0.002	0.098±0.003	0.226±0.006	0.175±0.001	0.054±0.001	0.098±0.002
	LIFT-INSDIF	0.066±0.001	0.066±0.002	0.091±0.002	**0.163±0.001**	**0.052±0.001**	0.064±0.001
	LIFT	**0.048±0.001**	**0.050±0.002**	**0.082±0.003**	0.165±0.002	0.057±0.001	**0.051±0.001**
	LIFT-MDDM	0.064±0.002	0.065±0.002	0.104±0.002	0.171±0.002	0.060±0.001	0.060±0.003
	LIFT-MLF	0.076±0.006	0.074±0.002	0.191±0.006	0.191±0.005	0.202±0.045	0.684±0.282

© 中国科学院软件研究所 http://www.jos.org.cn

Table 3 Predictive performance of each comparing algorithm (mean±std) on the six large-scale data sets (Continued)

表 3 对比算法在 6 个大规模数据集上的实验结果(mean±std)(续)

Criteria	Algorithms	rcv1 (subset1)	rcv1 (subset2)	bibtex	corel16k (sample1)	eurlex (subject matter)	tmc2007
One-Error ↓	BR	0.602±0.011	0.522±0.009	0.559±0.004	0.920±0.006	0.492±0.005	0.339±0.003
	ML-*k*NN	0.510±0.007	0.524±0.013	0.610±0.005	0.747±0.006	0.339±0.004	0.320±0.004
	LIFT-INSDIF	0.470±0.011	0.470±0.008	0.413±0.003	0.726±0.007	0.345±0.005	0.248±0.003
	LIFT	**0.412±0.008**	**0.421±0.009**	**0.407±0.008**	**0.691±0.008**	**0.329±0.006**	**0.213±0.004**
	LIFT-MDDM	0.430±0.009	0.431±0.007	0.416±0.004	0.713±0.015	0.370±0.005	0.234±0.006
	LIFT-MLF	0.467±0.015	0.464±0.010	0.545±0.008	0.715±0.055	0.732±0.064	0.511±0.185
Coverage ↓	BR	0.448±0.005	0.383±0.006	0.461±0.006	0.673±0.002	0.639±0.003	0.380±0.003
	ML-*k*NN	0.188±0.003	0.201±0.004	0.365±0.009	0.339±0.003	**0.092±0.001**	0.195±0.002
	LIFT-INSDIF	0.150±0.001	0.145±0.003	0.167±0.002	**0.318±0.003**	0.093±0.002	0.148±0.001
	LIFT	**0.120±0.002**	**0.123±0.005**	**0.149±0.006**	0.326±0.003	0.101±0.002	**0.129±0.001**
	LIFT-MDDM	0.155±0.004	0.147±0.002	0.190±0.004	0.329±0.003	0.106±0.002	0.142±0.004
	LIFT-MLF	0.183±0.011	0.173±0.004	0.318±0.008	0.385±0.008	0.321±0.091	0.683±0.146
Average precision ↑	BR	0.383±0.007	0.434±0.005	0.363±0.004	0.085±0.002	0.358±0.003	0.643±0.002
	ML-*k*NN	0.513±0.003	0.515±0.008	0.327±0.006	0.279±0.002	**0.650±0.003**	0.712±0.003
	LIFT-INSDIF	0.537±0.006	0.562±0.004	0.526±0.004	0.283±0.004	0.606±0.007	0.781±0.002
	LIFT	**0.605±0.003**	**0.617±0.006**	**0.542±0.008**	**0.311±0.003**	0.626±0.004	**0.814±0.002**
	LIFT-MDDM	0.580±0.004	0.600±0.004	0.518±0.004	0.289±0.005	0.580±0.005	0.795±0.004
	LIFT-MLF	0.544±0.009	0.566±0.006	0.376±0.007	0.301±0.018	0.167±0.013	0.222±0.091
Macro-Avg AUC ↑	BR	0.609±0.003	0.599±0.004	0.624±0.002	0.516±0.001	0.558±0.002	0.724±0.002
	ML-*k*NN	0.727±0.007	0.698±0.005	0.654±0.007	0.566±0.003	0.765±0.005	0.785±0.003
	LIFT-INSDIF	0.862±0.007	0.854±0.008	0.868±0.003	0.653±0.006	**0.855±0.005**	0.873±0.001
	LIFT	**0.917±0.004**	**0.901±0.007**	**0.902±0.004**	**0.695±0.002**	0.837±0.010	**0.908±0.001**
	LIFT-MDDM	0.859±0.006	0.840±0.003	0.847±0.003	0.657±0.003	0.828±0.013	0.890±0.003
	LIFT-MLF	0.801±0.007	0.791±0.013	0.749±0.006	0.573±0.007	0.500±0.001	0.503±0.013

Table 4(a) Paired *T*-test results (win/tie/lose) of comparing BR against algorithms with label-specific features mechanism in terms of each criteria on the total twelve data sets at significance level α=0.05

表 4(a) 对于每个评价指标,BR 与使用类属属性机制算法在 12 个数据集上的成对 *T* 检验(α=0.05)结果(win/tie/lose)

Criteria	BR versus			
	LIFT-INSDIF	LIFT	LIFT-MDDM	LIFT-MLF
Hamming loss	1/2/9	0/0/12	3/0/9	7/1/4
Ranking loss	0/0/12	0/0/12	0/0/12	3/1/8
One-Error	0/0/12	0/0/12	0/1/11	6/0/6
Coverage	0/0/12	0/0/12	0/0/12	3/1/8
Average precision	0/0/12	0/0/12	0/0/12	6/0/6
Macro AUC	0/0/12	0/0/12	0/0/12	5/0/7

参考第 1 组实验,本文使用成对 *T* 检验(检验水平α=0.05)对 LIFT 及其 3 种变体算法在每个数据集上的实验结果进行统计分析(见表 4(b)).从表 4(b)中可以看出:

(a) 在绝大多数情况下,LIFT 的分类性能优于其他 3 种变体算法——LIFT-INSDIF,LIFT-MDDM 以及 LIFT-MLF.在很少数据集的个别评价指标上,LIFT 的性能差于 LIFT-INSDIF 以及 LIFT-MDDM;

(b) 除了评价指标 One-Error 及 Coverage,LIFT-INSDIF 与 LIFT-MDDM 在其余 4 个评价指标上的结果相当.LIFT 与 LIFT-INSDIF 的性能优于或至少相当于 LIFT-MDDM 的性能,这说明相比于通过属性选择赋予每个标记相同的类属属性(即公式(6)),为每个标记构造不同的类属属性(即公式(3)与公式(7))是一种更有效的属性转换机制;

(c) 与表 4(a)中得到的结论一致,在每个评价指标上,LIFT-MLF 的性能都要差于其余 3 种算法.可能的原因是,该算法使用近邻样本构造元级属性(即公式(8))的方式不能有效地构造标记的类属属性.

概括来说,表 4(b)中的成对 *T* 检验统计结果证明了 LIFT 中的属性转换机制能非常有效地构造类别标记的

© 中国科学院软件研究所 http://www.jos.org.cn

类属属性.因此,LIFT 的类属属性机制可以作为很多多标记学习算法的预处理(pre-processing)部分,以提高学习系统的学习性能.

Table 4(b) Paired *T*-test results (win/tie/lose) of comparing LIFT against its variants in terms of each criteria on the total twelve data sets at significance level α=0.05

表 4(b) 对于每个评价指标,LIFT 与其变体算法在 12 个数据集上的成对 *T* 检验(α=0.05)结果(win/tie/lose)

Criteria	LIFT versus			LIFT-INSDIF versus		
	LIFT-INSDIF	LIFT-MDDM	LIFT-MLF	LIFT	LIFT-MDDM	LIFT-MLF
Hamming loss	10/0/2	11/0/1	11/1/0	2/0/10	5/4/3	10/2/0
Ranking loss	10/0/2	12/0/0	12/0/0	2/0/10	7/0/5	12/0/0
One-Error	9/3/0	9/2/1	11/1/0	0/3/9	2/4/6	9/3/0
Coverage	7/2/3	11/1/0	12/0/0	3/2/7	9/1/2	12/0/0
Average precision	12/0/0	11/0/1	11/1/0	0/0/12	4/1/7	9/1/2
Macro AUC	11/0/1	12/0/0	12/0/0	1/0/11	5/3/4	12/0/0
Criteria	LIFT-MDDM versus			LIFT-MLF versus		
	LIFT	LIFT-INSDIF	LIFT-MLF	LIFT	LIFT-INSDIF	LIFT-MDDM
Hamming loss	1/0/11	3/4/5	10/2/0	0/1/11	0/2/10	0/2/10
Ranking loss	0/0/12	5/0/7	12/0/0	0/0/12	0/0/12	0/0/12
One-Error	1/2/9	6/4/2	11/1/0	0/1/11	0/3/9	0/1/11
Coverage	0/1/11	2/1/9	12/0/0	0/0/12	0/0/12	0/0/12
Average precision	1/0/11	7/1/4	11/0/1	0/1/11	2/1/9	1/0/11
Macro AUC	0/0/12	4/3/5	12/0/0	0/0/12	0/0/12	0/0/12

4 总 结

与许多关注标记空间的算法不同,本文考察在属性空间进行操作对多标记学习算法学习性能的影响.本文在 12 个数据集上进行了大量的实验,将 LIFT 及其变体算法与 BR 进行对比分析,结果显示:有效的类属属性构造,的确能较大程度地提高多标记学习算法的学习性能.同时,LIFT 与其变体算法的统计检验结果表明,LIFT 的属性转换机制可有效地构造类别标记的类属属性.

值得注意的是,本文中所有算法均忽略标记之间的相关性.未来,我们可以尝试将类属属性和标记之间的相关性相结合,一种直观的方式是,为与标记配对相关的分类器[25]设计对应的二阶类属属性.

References:

[1] Tsoumakas G, Katakis I, Vlahavas I. Mining Multi-Label Data. Data Mining and Knowledge Discovery Handbook. 2nd ed., Berlin: Springer-Verlag, 2010. 667−685. [doi: 10.1007/978-0-387-09823-4_34]

[2] McMallum A. Multi-Label text classification with a mixture model trained by EM. In: Proc. of the Working Notes of the AAAI'99 Workshop on Text Learning. Orlando, 1999.

[3] Schapire RE, Singer Y. Boostexter: A boosting-based system for text categorization. Machine Learning, 2000,39(2):135−168. [doi: 10.1023/A:1007649029923]

[4] Jiang JY, Tsai SC, Lee SJ. FSKNN: Multi-Label text categorization based on fuzzy similarity and *k* nearest neighbors. Expert Systems with Applications, 2012,39(3):2813−2821. [doi: 10.1016/j.eswa.2011.08.141]

[5] Boutell MR, Luo J, Shen X, Brown CM. Learning multi-label scene classification. Pattern Recognition, 2004,37(9):1757−1771. [doi: 10.1016/j.patcog.2004.03.009]

[6] Wang M, Zhou X, Chua TS. Automatic image annotation via local multi-label classification. In: Proc. of the 7th ACM Int'l Conf. on Image and Video Retrieval. Niagara Falls, 2008. 17−26. [doi: 10.1145/1386352.1386359]

[7] Bao BK, Ni B, Mu Y, Yan S. Efficient region-aware large graph construction towards scalable multi-label propagation. Pattern Recognition, 2011,44(3):598−606. [doi: 10.1016/j.patcog.2010.10.001]

[8] 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]

[9] Zhang ML. LIFT: Multi-label learning with label-specific features. In: Proc. of the 22nd Int'l Joint Conf. on Artificial Intelligence. Barcelona, 2011. 1609−1614. [doi: 10.5591/978-1-57735-516-8/IJCAI11-270]

© 中国科学院软件研究所 http://www.jos.org.cn

[10] Zhang ML, Zhang K. Multi-Label learning by exploiting label dependency. In: Proc. of the 16th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. 2010. 999−1008. [doi: 10.1145/1835804.1835930]

[11] Huang SJ, Zhou ZH. Multi-Label learning by exploiting label correlations locally. In: Proc. of the 26th AAAI Conf. on Artificial Intelligence. 2012. 949−955.

[12] Guo YH, Gu SC. Multi-Label classification using conditional dependency networks. In: Proc. of the 22nd Int'l Joint Conf. on Artificial Intelligence. 2011. 1300−1305. [doi: 10.5591/978-1-57735-516-8/IJCAI11-220]

[13] Zhang Y, Zhou ZH. Multi-Label dimensionality reduction via dependency maximization. ACM Trans. on Knowledge Discovery from Data, 2010,4(3):Article 14. [doi: 10.1145/1839490.1839495]

[14] Zhou ZH, Zhang ML, Huang SJ, Li YF. Multi-Instance multi-label learning. Artificial Intelligence, 2012,176(1):2291−2320. [doi: 10.1016/j.artint.2011.10.002]

[15] Zhang ML, Zhou ZH. Multi-Label learning by instance differentiation. In: Proc. of the 22nd AAAI Conf. on Artificial Intelligence. Vancouver, 2007. 669−674.

[16] Yang Y, Gopal S. Multilabel classification with meta-level features in a learning-to-rank framework. Machine Learning, 2012, 88(1-2):47−68.

[17] Read J, Pfahringer B, Helms G, Frank E. Classifier chains for multi-label classification. Machine Learning, 2011,85(3):333−359. [doi: 10.1007/s10994-011-5256-5]

[18] Yang Y, Pedeson JO. A comparative study on feature selection in text categorization. In: Proc. of the 14th Int'l Conf. on Machine Learning. 1997. 412−420.

[19] Ghamrawi N, McCallum A. Collective multi-label classification. In: Proc. of the 14th ACM Int'l Conf. on Information and Knowledge Management. 2005. 195−200. [doi: 10.1145/1099554.1099591]

[20] Godbole S, Sarawagi S. Discriminative methods for multi-labeled classification. In: Proc. of the Advances in Knowledge Discovery and Data Mining. LNCS 3056, 2004. 22−30. [doi: 10.1007/978-3-540-24775-3_5]

[21] Tsoumakas G, Vlahavas I. Random *k*-labelsets: An ensemble method for multilabel classification. In: Proc. of the Machine Learning: ECML 2007. LNCS 4701, 2007. 406−417. [doi: 10.1007/978-3-540-74958-5_38]

[22] Hanley JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. 1982, 143(1): 29−36.

[23] 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]

[24] Chang CC, Lin CJ. LIBSVM: A library for support vector machines. ACM Trans. on Intelligent Systems and Technology, 2011, 2(3):Article 27.

[25] Furnkranz J, Hullermeierm E, Mencia EL, Brinker K. Multilabel classification via calibrated label ranking. Machine Learning, 2008,73(2):133−153. [doi: 10.1007/s10994-008-5064-8]

吴磊(1988－),男,江苏徐州人,硕士生,主要研究领域为机器学习,数据挖掘.
E-mail: wul@seu.edu.cn

张敏灵(1979－),男,博士,副教授,CCF 会员,主要研究领域为机器学习,数据挖掘.
E-mail: zhangml@seu.edu.cn

© 中国科学院软件研究所 http://www.jos.org.cn