第56卷 第4期
2020年7月

南京大学学报(自然科学)
JOURNAL OF NANJING UNIVERSITY
(NATURAL SCIENCE)

Vol. 56, No. 4
July, 2020

DOI:10.13232/j.cnki.jnju.2020.04.010

融合标签结构依赖性的标签分布学习

黄雨婷[1],徐媛媛[1],张恒汝[1*],闵 帆[1,2]

(1.西南石油大学计算机科学学院,成都,610500;2.西南石油大学人工智能研究院,成都,610500)

摘 要:针对现有标签分布学习(Label Distribution Learning,LDL)算法较少考虑标签间关联性的问题,提出一种融合结构化标签依赖性的LDL算法.算法分为扩展、学习和恢复三个阶段:在扩展阶段,结合成对标签之间的关联性,构建结构化标签依赖性;在学习阶段,结合该依赖性,构建学习框架;在恢复阶段,利用最小二乘法求解超定方程组以预测标签分布.与七种常用的标签分布学习算法相比,在八个开放数据集上进行实验,提出的算法在Euclidean距离、Sørensen距离、Squard χ^2距离、Kullback-Leibler散度、Intersection相似度和Fidelity相似度六个主流评估指标上明显占优.

关键词:标签分布学习,标签扩展,标签恢复,标签结构依赖性,有限存储拟牛顿法

中图分类号:TP391 **文献标识码**:A

Label distribution learning by exploiting structural label dependency

Huang Yuting[1], Xu Yuanyuan[1], Zhang Hengru[1*], Min Fan[1,2]

(1. School of Computer Science, Southwest Petroleum University, Chengdu, 610500, China;
2. Institute for Artificial Intelligence, Southwest Petroleum University, Chengdu, 610500, China)

Abstract: Aiming at the problem that the existing Label Distribution Learning (LDL) algorithms rarely consider the correlation between labels, an LDL algorithm that combines structured label dependencies is proposed. The algorithm is divided into three stages: expansion, learning and recovery. In the expansion stage, the association between the pair of labels is combined to construct a structured label dependency. In the learning stage, combining this dependency, a learning framework is constructed. In the recovery stage, the least square method is used to solve the overdetermined equations to predict the label distribution. Compared with the seven popular label distribution learning algorithms, experiments are conducted on eight open datasets. Our method is obviously superior at Euclidean distance, Sørensen distance, Squard χ^2 distance, Kullback-Leibler divergence, Intersection similarity and Fidelity similarity.

Key words: label distribution learning, label expansion, label recovery, structural label dependencies, the limited-memory quasi-Newton method

标签分布学习(Label Distribution Learning,LDL)由Geng et al于2010年提出[1],它是多标签学习(Muti-Label Learning,MLL)的泛化[2-3].LDL用标签集和所有标签的表征程度构成的分布来描述实例[4-6],而MLL仅用标签集的部分标签来描述实例[7-9].Zhou et al[10]将文本情绪分析问题泛化到LDL中,提高了情绪捕捉的准确率.Zhang et al[11]将人群计数问题泛化到LDL中,提高了计数的准确率.

图1a为需要标记的一个示例图片[12],其完整

基金项目:国家自然科学基金(61902328),四川省科技厅应用基础研究(2019YJ0314),四川省青年科技创新研究团队项目(2019JDTD0017),四川省大学生创新创业训练计划(S20190615090),西南石油大学本科课程教学改革研究项目(X2018KZ077)

收稿日期:2020-06-20

*通讯联系人,E-mail:zhanghrswpu@163.com

标签集为{森林,海洋,沙漠,城市}. 图1b说明MLL利用“森林”和“海洋”两个标签来描述该图片. 图1c说明LDL利用这四个标签构成的分布来描述该图片. Geng[2]提出BFGS-LLD(Effective Quasi-Newton Label Distribution Learning)算法,使用KL(Kullback-Leibler)散度[13]处理分布数据,但未考虑标签的相关性. Zheng et al[14]提出LDL-SCL(Label Distribution Learning by Exploiting Sample Correlations Locally)算法,使用K-means探索本地实例之间的相关性. Zhou et al[10]提出EDL(Emotion Distribution Learning from Texts)算法,使用Plutchik情绪轮捕捉情绪标签之间的相关性. 后两种算法显著提高了模型对标签分布的预测能力.

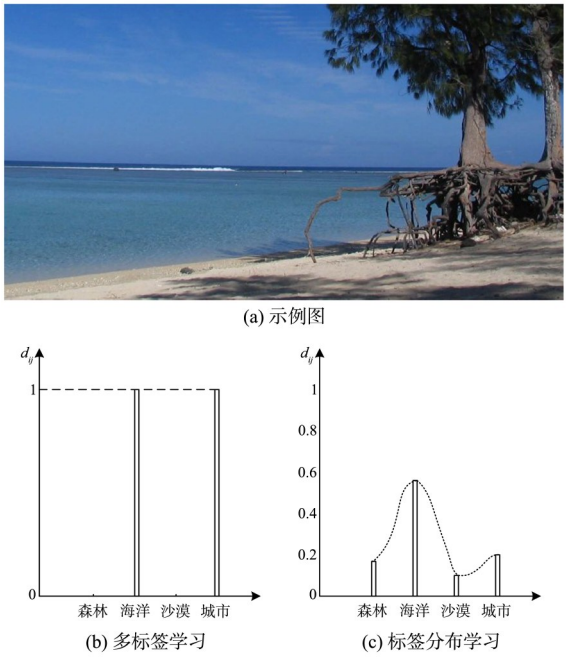

图1　MLL与LDL的比较

Fig.1　Difference between MLL and LDL

本文利用结构化标签依赖性来进行稳定准确的预测,其中依赖性通过标签表征度之间的差异来衡量. 本文的工作分为扩展、学习和恢复三个阶段. 在扩展阶段,将c个标签扩展为$c(c-1)/2$个,任意两个标签之差构成了新的标签,结合成对标签之间的关联性,构建结构化标签依赖性. 在学习阶段,通过使用有限存储拟牛顿法求解最优模型,并预测新标签对应的分布. 在恢复阶段,求解超定方程组以预测原始标签对应的标签分布.

在八个来自芽殖酿酒酵母的生物学实验的数据集上,将本文提出的算法与七种主流算法进行对比实验. 结果表明,在多数的评价指标中,本文的算法比七种已有的LDL算法表现更好.

1　相关工作

首先提出问题描述,再回顾已有的LDL算法. 表1列出本文中使用的符号.

表1　符号系统

Table 1　Notations

Notations	Meaning
$\mathbb{R}^q$	q-dimensional input space
Y	The complete set of labels
S	The training set
x_i	The i-th instance
d_i	The label distribution associated with x_i
p_i	The predicted label distribution associated with x_i
x_{ir}	The r-th feature of x_i
d_{ij}	The description degree of the j-th label to x_i
θ	The distance-mapping model
X	The instance matrix
D	The label distribution matrix

1.1　问题描述　与单标签和多标签学习相比,标签分布学习以一种更自然的方式去标记实例,并且为它的每个可能的标签分配一个数值.

下面给出它的形式化定义[2]. 令$\boldsymbol{X}=\mathbb{R}^q$为$q$维的输入空间,表示特征矩阵. $\boldsymbol{Y}=\{y_1,y_2,\cdots,y_c\}$为完整标签集,$\boldsymbol{D}$表示标签分布矩阵. 给定一个训练集$S=\{X,D\}=\{\{x_1,d_1\},\{x_2,d_2\},\cdots,\{x_n,d_n\}\}$,其中$x_i=[x_{i1},x_{i2},\cdots,x_{iq}]\in\boldsymbol{X}$为第$i$个实例,$d_i=[d_{i1},d_{i2},\cdots,d_{ic}]\in[0,1]^c$为$x_i$对应的实际标签分布,$d_{ij}$是标签$y_j$对$x_i$的描述度,且$\sum_{j=1}^{c}d_{ij}=1$.

当前,LDL的目标是学习到一个可以描述$\boldsymbol{X}$和$\boldsymbol{D}$之间关系的模型. 预测的分布矩阵$\boldsymbol{P}=[p_1,p_2,\cdots,p_c]$由该模型和$\boldsymbol{X}$计算得到,其中$p_i=[p_{i1},p_{i2},\cdots,p_{ic}]$,$p_{ij}$是标签$y_j$对$x_i$的预测表征度.

1.2 已有的LDL算法 在已有的LDL算法中，有三种算法构造策略[14].

(1)问题转换策略：将LDL问题转换为单标签问题或多标签问题再处理. 例如PT-SVM(Problem Transformation Support Vector Machine)算法[2]将标签分布数据转换为加权的单标签数据后，再使用支持向量机的方法处理数据.

(2)算法适应策略：扩展现有的学习算法以处理标签分布. 例如AA-BP(Algorithm Adaptation Backpropagation)算法[2]构建了一个三层反向传播神经网络处理标签分布，为保证输出数据满足分布特征，使用Softmax函数作为激活函数.

(3)特殊算法策略：根据LDL的特征，设计特殊的专用算法直接匹配LDL问题. 例如BFGS-LLD(Effective Quasi-Newton Label Distribution Learning)算法[2]根据分布数据的特征，学习到一个模型来直接预测标签分布.

通常，特殊算法由三部分组成：输出模型、目标函数和优化方法.

(1)输出模型：已有的LDL算法主要使用最大熵模型[15]作为输出模型，例如IIS-LLD(Improved Iterative Scaling Label Distribution Learning)算法[2]、LDL-SCL算法[14]、LDL4C(Label Distribution Learning Algorithm for Classification)算法[16]、EDL算法[10]和LDLLC(Label Distribution Learning by Exploiting Label Correlations)算法[17].

(2)目标函数：已有的LDL算法使用多种相似度公式来构建目标函数，以衡量原始分布和预测分布之间差异. 例如Geng[2]使用KL散度，Xu and Geng[18]使用离散的Jeffery散度.

(3)优化方法：已有的LDL算法通常使用多种优化算法以求解获得最优模型，例如改进迭代缩放(Improved Iterative Scaling, IIS)方法[2]、有效拟牛顿(Effective Quasi-Newton, BFGS)方法[2]、交替方向乘子方法(Alternating Direction Method of Multipliers, ADMM)[19-20]和有限存储拟牛顿法(Limited-memory Quasi-Newton, L-BFGS)[10,17].

2 本文的SD-LDL算法

本文的SD-LDL算法分为三个阶段：标签扩展、标签学习和标签恢复.

2.1 标签扩展 现有的LDL算法通常都独立地考虑每个标签，未考虑标签之间的关系. 受MLL中考虑标签的成对相关性[21]的启发，本文提出以下算法来获取标签之间的结构依赖性. 对于实例x_i，标签y_j与标签y_k对它的表征度的差异为：

$$\delta_{is'}=d_{ij}-d_{ik}=p\left(y_j\middle|x_i\right)-p\left(y_k\middle|x\right) \tag{1}$$

其中，$1\leqslant j<k\leqslant c$，且

$$s'=s(j,k)=\frac{(j-1)(2c-j)}{2}+(k-j) \tag{2}$$

是原始标签对(j,k)对应的新标签索引.

通过上述方式，获得与实例x_i相关联的新标签分布$\delta_i=\left[\delta_{i1},\delta_{i2},\cdots,\delta_{ic'}\right]$，该分布能够表征实例$x_i$的标签结构依赖性，且原始的$c$个标签被扩展为$c(c-1)/2$个新标签. 为便于表示，令$c'=c(c-1)/2$.

2.2 标签学习 遵循特殊算法策略的构建结构，本文的学习算法同样由三部分构成：输出模型、目标函数和优化方法.

(1)输出模型. 经2.1标签扩展后，特征矩阵$\boldsymbol{X}$仍然为q维，标签分布矩阵$\boldsymbol{D}$扩展为c'维. 构建一个$q\times c'$的模型θ表示$\boldsymbol{X}$与$\boldsymbol{D}$的映射关系，通过该模型和$\boldsymbol{X}$可计算得到与实例x_i相关联的新标签的预测分布$\hat{\delta}_i=\left[\hat{\delta}_{i1},\hat{\delta}_{i2},\cdots,\hat{\delta}_{ic'}\right]$. 第$s'$个新标签对实例$x_i$的预测表征度为：

$$\hat{\delta}_{is'}=p\left(\delta_{is'}\middle|x_i;\theta\right)=\sum_{r=1}^{q}x_{ir}\theta_{rs'} \tag{3}$$

其中

$$\hat{\delta}_{is'}=p_{ij}-p_{ik} \tag{4}$$

p_{ij}和p_{ik}分别是第j个原始标签和第k个原始标签对实例x_i的预测表征度.

(2)目标函数. 采用KL散度[13]设计目标函数：

$$T(\theta)=KL\left(\delta_i\middle\|\hat{\delta}_i\right)=\sum_{i=1}^{n}\sum_{s'=1}^{c'}\left(\delta_{is'}\ln\frac{\delta_{is'}}{\hat{\delta}_{is}}\right)+\lambda\|\theta\|_F^2 \tag{5}$$

其中,$\|\cdot\|_F$为F范数,λ为平衡因子参数,添加该项以避免过拟合.

(3)优化方法.本文采用L-BFGS方法[22]作为优化方法以获取最优模型θ.L-BFGS方法是在BFGS方法的基础上,对拟牛顿法的进一步精简.在拟牛顿法中,需要存储每一轮迭代产生的Hessian矩阵,这对机器的存储能力以及算法的时间复杂度都是很大的挑战.BFGS方法的基本思想是使用一个近似矩阵去拟合Hessian矩阵,以避免完整Hessian矩阵的计算及存储.L-BFGS方法仅存储部分用于拟合Hessian矩阵的向量,在保证拟合效果的前提下,达到进一步的存储空间的节约.有关L-BFGS方法的更多信息,请参阅附录.

2.3　标签恢复　在标签恢复阶段,通过标签学习获得的新标签对应的预测分布恢复为原始标签对应的预测分布.

根据式(4),对与实例x_i相关联的新标签的预测分布进行归一化,可以得到以下方程组:

$$\begin{cases}\hat{\delta}_{i1}=p_{i2}-p_{i1}\\ \hat{\delta}_{i2}=p_{i3}-p_{i1}\\ \vdots\\ \hat{\delta}_{ic'}=p_{ic}-p_{i(c-1)}\\ \sum\limits_{j=1}^{c}p_{ij}=1\end{cases}\tag{6}$$

该方程组由c个变量和$\frac{c(c-1)}{2}+1$个方程组成,求解上述方程组,获得原始标签对应的预测分布.$0<c<3$时$\frac{c(c-1)}{2}+1=c$,它是正定方程组且有唯一解.$c>3$时$\frac{c(c-1)}{2}+1>c$,它是超定方程组且有多个解,采用最小二乘法解决.

3　实　验

本节将SD-LDL算法与PT-Bayes[2],PT-SVM[2,16],AA-kNN[2,17],AA-BP[2],IIS-LLD[2,14],BFGS-LLD[2]和EDL[10]七种主流的LDL算法进行了比较.

3.1　数据集　表2列出了从芽殖酵母的八个生物学实验中收集得到的八个真实数据集[23].实例为2465个酵母基因,特征为长度24的系统发育谱图,标签为不同生物实验中的离散时间点,数量范围为4至18.

表2　本文采用的数据集

Table 2　Datasets used in this paper

Dataset	Instance	Feature	Label
Alpha	2465	24	18
Cdc	2465	24	15
Elu	2465	24	14
Diau	2465	24	7
Heat	2465	24	6
Spo	2465	24	6
Cold	2465	24	4
Dtt	2465	24	4

3.2　评价指标　表3列出了评估LDL算法的六个评价指标.“↓”表示“越小越好”,“↑”表示“越大越好”.

表3　LDL算法的评价指标

Table 3　Evaluation measures for LDL algorithms

Measure	Formula
Euclidean[24] ↓	$dis=\sqrt{\sum\limits_{j=1}^{c}(p_j-q_j)^2}$
Sørensen[25] ↓	$dis=\dfrac{\sum\limits_{j=1}^{c}\lvert p_j-q_j\rvert}{\sum\limits_{j=1}^{c}\lvert p_j+q_j\rvert}$
Squard χ^2[26] ↓	$dis=\sum\limits_{j=1}^{c}\dfrac{(p_j-q_j)^2}{p_j+q_j}$
Kullback-Leibler (KL)[13] ↓	$dis=\sum\limits_{j=1}^{c}p_i\ln\dfrac{p_i}{d_i}$
Intersection[27] ↑	$sim=\min(p_j-q_j)$
Fidelity[28] ↑	$sim=\sum\limits_{j=1}^{c}\sqrt{p_jq_j}$

3.3　参数设置　表4列出SD-LDL算法与七种主流LDL算法相应的参数设置.

对于SD-LDL,式(8)中的参数设置为$\lambda=0.1$.对于PT-Bayes,使用极大似然估计作为估计高斯类条件概率密度函数.对于PT-SVM,参数设置为$C=1.0$,$\Gamma=0.01$.对于AA-kNN,参数k是邻居的数量,设置为5.对于AA-BP,参数n是隐藏层神经元的数量,设置为60.对于BFGS-LLD,参数设置为$c_1=10^{-4}$,$c_2=0.9$.

表 4 参数设置

Table 4 Parameters settings

Algorithm	Settings
SD-LDL	$\lambda=0.1$
PT-Bayes	极大似然估计
PT-SVM	$C=1.0, \Gamma=0.01$
AA-kNN	$k=5$
AA-BP	$n=60$
IIS-LLD	—
BFGS-LLD	$c_1=10^{-4}, c_2=0.9$
EDL	—

3.4 实验结果 表 5 至表 12 列出 10 次实验的平均结果和标准差,括号中的数字表示当前方法性能的排名,黑体字表示排名第一的结果. 首先比较表中的平均结果,如果平均结果相同,再比较标准差,标准差越小,性能越好.

首先,对比 SD-LDL 算法与 EDL 算法,这两个算法均考虑了标签间的关系,但在所有数据集上的实验结果表明,本文提出的算法依然占优.

其次,对比 PT-Bayes 算法、PT-SVM 算法、AA-kNN 算法、AA-BP 算法、IIS-LLD 算法和 BFGS-LLD 算法,这六个算法均未考虑标签间的

表 5 Alpha 数据集上的实验结果

Table 5 Experimental results on the Alpha dataset

	Euclidean ↓	Sørensen ↓	Squard χ^2 ↓	KL ↓	Intersection ↑	Fidelity ↑
SD-LDL	**0.0236±0.0003(1)**	**0.0386±0.0003(1)**	**0.0057±0.0003(1)**	**0.0056±0.0002(1)**	**0.9614±0.0005(1)**	**0.9986±0.0003(1)**
PT-Bayes	0.2298±0.0124(8)	0.3485±0.0154(8)	0.3879±0.0277(8)	0.5607±0.0710(8)	0.6515±0.0154(8)	0.8777±0.0100(8)
PT-SVM	0.0276±0.0006(5)	0.0445±0.0009(5)	0.0071±0.0003(5)	0.0071±0.0003(5)	0.9565±0.0009(5)	0.9981±0.0001(5)
AA-kNN	0.0279±0.0006(6)	0.0449±0.0012(6)	0.0073±0.0003(6)	0.0074±0.0004(7)	0.9561±0.0012(6)	0.9980±0.0001(6)
AA-BP	0.0871±0.0070(7)	0.1475±0.0131(7)	0.1399±0.0501(7)	0.0073±0.0058(6)	0.8538±0.0117(7)	0.9839±0.0017(7)
IIS-LLD	0.269±0.0004(4)	0.0429±0.0012(3)	0.0069±0.0004(4)	0.0069±0.0004(4)	0.9571±0.0012(3)	0.9983±0.0011(4)
BFGS-LLD	0.0251±0.0004(2)	0.0408±0.0011(2)	0.0063±0.0008(2)	0.0063±0.0004(2)	0.9574±0.0009(2)	0.9985±0.0011(2)
EDL	0.0260±0.0011(3)	0.0429±0.0022(4)	0.0067±0.0006(3)	0.0068±0.0006(3)	0.9570±0.0022(4)	0.9983±0.0002(3)

表 6 Cdc 数据集上的实验结果

Table 6 Experimental results on the Cdc dataset

	Euclidean ↓	Sørensen ↓	Squard χ^2 ↓	KL ↓	Intersection ↑	Fidelity ↑
SD-LDL	**0.0282±0.0004(1)**	**0.0428±0.0008(1)**	**0.0073±0.0005(3)**	**0.0071±0.0001(2)**	**0.9572±0.0004(1)**	**0.9983±0.0003(1)**
PT-Bayes	0.2399±0.0103(8)	0.3455±0.0111(8)	3853±0.0210(8)	0.5374±0.0503(8)	0.6545±0.0111(8)	0.8778±0.0075(8)
PT-SVM	0.0298±0.0007(4)	0.0458±0.0012(5)	0.0077±0.0004(5)	0.0076±0.0004(5)	0.9554±0.0012(5)	0.9980±0.0001(5)
AA-kNN	0.0301±0.0009(6)	0.0462±0.0013(6)	0.0080±0.0004(6)	0.0079±0.0004(6)	0.9538±0.0013(6)	0.9980±0.0001(5)
AA-BP	0.0769±0.0081(7)	0.1192±0.0109(7)	0.0842±0.0281(7)	0.0511±0.0121(7)	0.8829±0.0134(7)	0.9879±0.0051(7)
IIS-LLD	0.0290±0.0010(5)	0.0445±0.0015(3)	0.0073±0.0005(4)	0.0072±0.0005(4)	0.9556±0.0015(4)	0.9982±0.0012(4)
BFGS-LLD	0.0284±0.0011(3)	0.0449±0.0016(4)	**0.0070±0.0004(1)**	**0.0070±0.0005(1)**	0.9558±0.0016(3)	0.9983±0.0011(2)
EDL	0.0283±0.0006(2)	0.0429±0.0008(2)	0.0072±0.0004(2)	0.0072±0.0004(3)	0.9571±0.0008(2)	0.9982±0.0001(3)

表 7 Elu 数据集上的实验结果

Table 7 Experimental results on the Elu dataset

	Euclidean ↓	Sørensen ↓	Squard χ^2 ↓	KL ↓	Intersection ↑	Fidelity ↑
SD-LDL	**0.0282±0.0004(1)**	**0.0423±0.0006(1)**	**0.0064±0.0005(1)**	**0.0063±0.0003(1)**	**0.9577±0.0006(1)**	**0.9984±0.0003(1)**
PT-Bayes	0.2588±0.0203(8)	0.3558±0.0198(8)	0.4081±0.0408(8)	0.6062±0.1030(8)	0.6442±0.0198(8)	0.8689±0.0156(8)
PT-SVM	0.0293±0.0008(3)	0.0438±0.0012(3)	0.0068±0.0005(3)	0.0068±0.0005(3)	0.9562±0.0012(3)	0.9983±0.0002(3)
AA-kNN	0.0297±0.0010(4)	0.0443±0.0014(4)	0.0071±0.0006(5)	0.0071±0.0006(5)	0.9557±0.0014(4)	0.9982±0.0002(4)
AA-BP	0.0733±0.0037(7)	0.1100±0.0048(7)	0.0731±0.0026(7)	0.0481±0.0061(7)	0.8891±0.0064(7)	0.9890±0.0025(7)
IIS-LLD	0.0307±0.0009(5)	0.0472±0.0014(5)	0.0071±0.0004(4)	0.0071±0.0004(4)	0.9528±0.0015(6)	0.9982±0.0035(5)
BFGS-LLD	0.0308±0.0009(6)	0.0475±0.0012(6)	0.0075±0.0004(6)	0.0073±0.0003(6)	0.9552±0.0017(5)	0.9979±0.0009(6)
EDL	0.0289±0.0005(2)	0.0431±0.0008(2)	0.0067±0.0003(2)	0.0067±0.0003(2)	0.9569±0.0007(2)	0.9983±0.0001(2)

表8　Diau数据集上的实验结果

Table 8　Experimental results on the Diau dataset

	Euclidean ↓	Sørensen ↓	Squard χ^2 ↓	KL ↓	Intersection ↑	Fidelity ↑
SD-LDL	0.0602±0.0009(5)	0.0660±0.0009(5)	0.0160±0.0015(5)	0.0155±0.0011(5)	0.9340±0.0009(5)	0.9959±0.0008(5)
PT-Bayes	0.4027±0.0183(8)	0.4177±0.0170(8)	0.5280±0.0281(8)	0.8512±0.0772(8)	0.5823±0.0170(8)	0.8230±0.0107(8)
PT-SVM	0.0628±0.0037(6)	0.0686±0.0041(6)	0.0169±0.0018(6)	0.0167±0.0017(6)	0.9314±0.0041(6)	0.9957±0.0004(6)
AA-kNN	0.0567±0.0019(3)	0.0622±0.0022(3)	0.0145±0.0011(3)	0.0145±0.0010(3)	0.9378±0.0022(3)	0.9963±0.0003(3)
AA-BP	0.0802±0.0051(7)	0.0863±0.0059(7)	0.0276±0.0013(7)	0.0291±0.0069(7)	0.9142±0.0067(7)	0.9929±0.0031(7)
IIS-LLD	0.0539±0.0031(2)	0.0593±0.0032(2)	0.0144±0.0014(2)	0.0141±0.0013(2)	0.9407±0.0003(2)	0.9964±0.0036(2)
BFGS-LLD	**0.0444±0.0022(1)**	**0.0476±0.0023(1)**	**0.0089±0.0008(1)**	**0.0083±0.0009(1)**	**0.9513±0.0027(1)**	**0.9978±0.0031(1)**
EDL	0.0597±0.0010(4)	0.0653±0.0010(4)	0.0158±0.0005(4)	0.0155±0.0005(4)	0.9347±0.0010(4)	0.9960±0.0002(4)

表9　Heat数据集上的实验结果

Table 9　Experimental results on the Heat dataset

	Euclidean ↓	Sørensen ↓	Squard χ^2 ↓	KL ↓	Intersection ↑	Fidelity ↑
SD-LDL	**0.0602±0.0009(1)**	**0.0607±0.0008(1)**	**0.0131±0.0012(1)**	**0.0130±0.0008(1)**	**0.9393±0.0008(1)**	**0.9967±0.0008(1)**
PT-Bayes	0.4500±0.0231(8)	0.4354±0.0193(8)	0.5450±0.0361(8)	0.8678±0.1198(8)	0.5646±0.0193(8)	0.8180±0.0131(8)
PT-SVM	0.0625±0.0023(3)	0.0627±0.0022(2)	0.0141±0.0010(2)	0.0141±0.0010(2)	0.9373±0.0022(3)	0.9964±0.0003(2)
AA-kNN	0.0624±0.0020(2)	0.0632±0.0018(3)	0.0141±0.0010(2)	0.0141±0.0010(2)	0.9368±0.0018(2)	0.9964±0.0003(2)
AA-BP	0.0793±0.0068(7)	0.0822±0.0071(7)	0.0235±0.0047(7)	0.0246±0.0053(7)	0.9198±0.0061(7)	0.9937±0.0028(7)
IIS-LLD	0.0703±0.0036(5)	0.0692±0.0033(5)	0.0182±0.0016(5)	0.0182±0.0016(5)	0.9309±0.0033(5)	0.9954±0.0042(6)
BFGS-LLD	0.0728±0.0031(6)	0.0791±0.0029(6)	0.0188±0.0016(6)	0.0186±0.0015(6)	0.9304±0.0034(6)	0.9961±0.0048(5)
EDL	0.0629±0.0016(4)	0.0633±0.0017(4)	0.0143±0.0008(4)	0.0143±0.0008(4)	0.9366±0.0017(4)	0.9963±0.0003(4)

表10　Spo数据集上的实验结果

Table 10　Experimental results on the Spo dataset

	Euclidean ↓	Sørensen ↓	Squard χ^2 ↓	KL ↓	Intersection ↑	Fidelity ↑
SD-LDL	0.0835±0.0012(2)	0.0860±0.0011(2)	0.0265±0.0014(3)	0.0266±0.0011(3)	0.9140±0.0011(3)	0.9932±0.0006(4)
PT-Bayes	0.4038±0.0162(8)	0.4030±0.0134(8)	0.4972±0.0246(8)	0.7172±0.0840(8)	0.5971±0.0134(8)	0.8342±0.0095(8)
PT-SVM	0.0878±0.0019(5)	0.0893±0.0022(5)	0.0280±0.0015(5)	0.0284±0.0015(5)	0.9107±0.0022(5)	0.9929±0.0004(5)
AA-kNN	0.0879±0.0030(6)	0.0899±0.0024(6)	0.0286±0.0020(6)	0.0286±0.0002(6)	0.9096±0.0034(6)	0.9927±0.0005(6)
AA-BP	0.0979±0.0041(7)	0.1012±0.0038(7)	0.0344±0.0038(7)	0.0359±0.0039(7)	0.8982±0.0037(7)	0.9906±0.0010(7)
IIS-LLD	0.0863±0.0041(4)	0.0861±0.0036(3)	0.0251±0.0036(2)	0.0252±0.0022(2)	0.9139±0.0036(3)	0.9937±0.0005(2)
BFGS-LLD	**0.0819±0.0045(1)**	**0.0833±0.0038(1)**	**0.0229±0.0019(1)**	**0.0226±0.0021(1)**	**0.9168±0.0039(1)**	**0.9951±0.0007(1)**
EDL	0.0843±0.0029(3)	0.0872±0.0029(4)	0.0268±0.0015(4)	0.0269±0.0016(4)	0.9128±0.0028(4)	0.9932±0.0004(3)

表11　Cold数据集上的实验结果

Table 11　Experimental results on the Cold dataset

	Euclidean ↓	Sørensen ↓	Squard χ^2 ↓	KL ↓	Intersection ↑	Fidelity ↑
SD-LDL	**0.0713±0.0009(1)**	**0.0619±0.0009(1)**	**0.0133±0.0019(1)**	**0.0130±0.0014(1)**	**0.9381±0.0009(1)**	**0.9968±0.0014(1)**
PT-Bayes	0.5252±0.0224(8)	0.4479±0.0189(8)	0.5873±0.0352(8)	0.9089±0.1042(8)	0.5521±0.0189(8)	0.7991±0.0134(8)
PT-SVM	0.0753±0.0080(4)	0.0654±0.0069(5)	0.0147±0.0033(4)	0.0146±0.0033(4)	0.9346±0.0069(5)	0.9963±0.0008(4)
AA-kNN	0.0724±0.0027(2)	0.0630±0.0024(2)	0.0136±0.0011(2)	0.0136±0.0011(2)	0.9370±0.0024(2)	0.9966±0.0003(3)
AA-BP	0.0838±0.0045(7)	0.0710±0.0027(7)	0178±0.0011(7)	0.0163±0.0030(7)	0.9328±0.0029(7)	0.9952±0.0017(7)
IIS-LLD	0.0767±0.0004(5)	0.0653±0.0034(4)	0.0157±0.0015(6)	0.0155±0.0015(6)	0.9347±0.0034(4)	0.9960±0.0039(6)
BFGS-LLD	0.0745±0.0004(3)	0.0641±0.0035(3)	0.0139±0.0013(3)	0.0143±0.0015(3)	0.9348±0.0035(3)	0.9968±0.0036(2)
EDL	0.0771±0.0018(6)	0.0668±0.0016(6)	0.0154±0.0009(5)	0.0153±0.0009(5)	0.9332±0.0016(6)	0.9961±0.0003(5)

表 12 Dtt数据集上的实验结果

Table 12 Experimental results on the Dtt dataset

	Euclidean ↓	Sørensen ↓	Squard χ^2 ↓	KL ↓	Intersection ↑	Fidelity ↑
SD-LDL	**0.0495±0.0015(1)**	0.0429±0.0013(2)	0.0066±0.0038(2)	0.0063±0.0029(2)	0.9571±0.0013(2)	0.9983±0.0021(3)
PT-Bayes	0.4879±0.0242(8)	0.4156±0.0192(8)	0.5416±0.0438(8)	0.9069±0.1580(8)	0.5844±0.0192(8)	0.8113±0.0186(8)
PT-SVM	0.0516±0.0029(5)	0.0447±0.0024(5)	0.0071±0.0009(6)	0.0071±0.0009(6)	0.9553±0.0024(5)	0.9982±0.0003(5)
AA-kNN	0.0512±0.0019(4)	0.0443±0.0017(4)	0.0071±0.0007(5)	0.0070±0.0007(5)	0.9557±0.0017(4)	0.9982±0.0002(4)
AA-BP	0.0622±0.0032(7)	0.0531±0.0029(7)	0.0097±0.0012(7)	0.0122±0.0037(7)	0.9465±0.0024(7)	0.9969±0.0011(7)
IIS-LLD	0.0535±0.0023(6)	0.0480±0.0023(6)	0.0068±0.0005(3)	0.0068±0.0005(3)	0.9520±0.0023(6)	0.9983±0.0013(2)
BFGS-LLD	0.0495±0.0019(2)	**0.0409±0.0017(1)**	**0.0058±0.0005(1)**	**0.0054±0.0004(1)**	**0.9584±0.0023(1)**	**0.9989±0.0010(1)**
EDL	0.0508±0.0022(3)	0.0440±0.0018(3)	0.0069±0.0007(4)	0.0068±0.0008(4)	0.9560±0.0018(3)	0.9982±0.0003(5)

关系.BFGS-LLD算法在五个数据集上的表现占优,AA-kNN算法仅在两个数据集上的表现占优,IIS-LLD算法和PT-SVM算法仅在一个数据集上的表现较为占优.AA-BP算法和PT-Bayes算法在所有数据集上的表现均不佳.

最后,将本文提出的SD-LDL算法与七种主流的LDL算法作对比.本文提出的SD-LDL算法在七个数据集上的表现均占优.

3.5 讨 论 受MLL中考虑标签的成对相关性的启发,本文通过获取标签间的结构依赖性以提高模型的预测能力.通过标签扩展获得该依赖性,使得预测模型θ的维度增加,能够更细粒度地表征特征矩阵$\boldsymbol{X}$和标签分布矩阵$\boldsymbol{D}$之间的关系.实验结果证明在多数数据集上,本文提出的SD-LDL算法的表现比七种主流的LDL算法更好.

对于Alpha,Cold,Elu,Cdc和Heat数据集,本文提出的SD-LDL算法在绝大多数的评价指标上均为最佳.这些数据集拥有的标签数量较多,有更多可供利用的标签结构依赖性.

对于Spo和Diau数据集,SD-LDL算法和EDL算法的表现均略逊色于BFGS-LLD算法.前两种算法均考虑了标签间的关系,而BFGS-LLD算法并未考虑.可能是由于这两个数据集中标签间的关系不强,或者是由于现有算法中用于表征标签间关系的方式不适用于这两个数据集.

4 结 论

LDL是一个通用的学习框架,它可以有效地处理标签歧义问题.为了利用成对标签的结构依赖性,本文提出了SD-LDL算法.本文的方法包括标签扩展、标签学习和标签恢复三个阶段.实验结果表明,该算法适用于标签分布框架,并且比大多数现有的LDL算法表现更好.

在未来的工作中,将尝试从以下几个方面提高LDL算法的性能:

(1)在扩展阶段,利用新的差异函数来表示结构标签依赖性.

(2)在学习阶段,采用属性约简以减少时间复杂度,并且使用其他衡量相似度的函数.

(3)在恢复阶段,探索新的标签恢复方案.

参考文献

[1] Geng X, Smith-Miles K, Zhou Z H. Facial age estimation by learning from label distributions//Proceedings of the 24th AAAI Conference on Artificial Intelligence. Atlanta, GA, USA: AAAI, 2010: 451-456.

[2] Geng X. Label distribution learning. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(7): 1734-1748.

[3] Yang X, Gao B B, Xing C, et al. Deep label distribution learning for apparent age estimation// Proceedings of the 2015 IEEE International Conference on Computer Vision Workshops. Santiago, Chile: IEEE, 2015: 102-108.

[4] Geng X, Ling M G. Soft video parsing by label distribution learning//Proceedings of the 31th AAAI Conference on Artificial Intelligence. San Francisco, CA, USA: AAAI Press, 2017: 1331-1337.

[5] Geng X, Wang Q, Xia Y. Facial age estimation by adaptive label distribution learning//Proceedings of the 22nd International Conference on Pattern Recognition. Stockholm, Sweden: IEEE, 2014: 4465-4470.

[6] Geng X, Yin C, Zhou Z H. Facial age estimation by learning from label distributions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(10):2401－2412.

[7] Zhang M L, Zhang K. Multi‐label learning by exploiting label dependency//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington DC, USA:ACM, 2010:999－1008.

[8] Wei B, Kwok J T Y. Multilabel classification with label correlations and missing labels//Proceedings of the 28th AAAI Conference on Artificial Intelligence. Québec City, Canada, 2014:1680－1686.

[9] Huang S J, Zhou Z H. Multi//label learning by exploiting label correlations locally//Proceedings of the 26th AAAI Conference on Artificial Intelligence. Toronto, Canada:AAAI, 2012:949－955.

[10] Zhou D Y, Zhang X, Zhou Y, et al. Emotion distribution learning from texts//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, TX, USA: Association for Computational Linguistics, 2016:638－647.

[11] Zhang Z X, Wang M, Geng X. Crowd counting in public video surveillance by label distribution learning. Neurocomputing, 2015, 166:151－163.

[12] Jégou H, Douze M, Schmid C. Hamming embedding and weak geometric consistency for large scale image search//Proceedings of the 10th European Conference on Computer Vision. Springer Berlin Heidelberg, 2008:304－317.

[13] Kullback S, Leibler R A. On information and sufficiency. Annals of Mathematical Statistics, 1951, 22(1):79－86.

[14] Zheng X, Jia X Y, Li W W. Label distribution learning by exploiting sample correlations locally// Proceedings of the 32nd AAAI Conference on Artificial Intelligence. New Orleans, LA, USA: AAAI, 2018:4556－4563.

[15] Berger A L, Pietra V J D, Pietra S A D. A maximum entropy approach to natural language processing. Computational Linguistics, 1996, 22(1):39－71.

[16] Wang J, Geng X. Classification with label distribution learning//Proceedings of the 28th International Joint Conference on Artificial Intelligence. Macao, China: International Joint Conferences on Artificial Intelligence Organization, 2019:3712－3718.

[17] Jia X Y, Li W W, Liu J Y, et al. Label distribution learning by exploiting label correlations//Proceedings of the 32nd AAAI Conference on Artificial Intelligence. New Orleans, LA, USA: AAAI, 2018: 3310－3317.

[18] Xu C D, Geng X. Hierarchical classification based on label distribution learning//Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Honolulu, HI, USA:AAAI, 2019:5533－5540.

[19] Ren T T, Jia X Y, Li W W, et al. Label distribution learning with label correlations via low‐rank approximation//Proceedings of the 28th International Joint Conference on Artificial Intelligence. Macao, China: International Joint Conferences on Artificial Intelligence Organization, 2019:3325－3331.

[20] Ren T T, Jia X Y, Li W W, et al. Label distribution learning with label‐specific features//Proceedings of the 28th International Joint Conference on Artificial Intelligence. Macao, China: International Joint Conferences on Artificial Intelligence Organization, 2019:3318－3324.

[21] Mencía E L, Park S H, Fürnkranz J. Efficient voting prediction for pairwise multilabel classification. Neurocomputing, 2010, 73(7－9):1164－1176.

[22] Yuan Y X. A modified BFGS algorithm for unconstrained optimization. IMA Journal of Numerical Analysis, 1991, 11(3):325－332.

[23] Eisen M B, Spellman P T, Brown P O, et al. Cluster analysis and display of genome‐wide expression patterns. The National Academy of Sciences of the United States of America, 1998, 95(25): 14863－14868.

[24] Danielsson P E. Euclidean distance mapping. Computer Graphics and Image Processing, 1980, 14 (3):227－248.

[25] Sørenson T. A method of establishing groups of equal amplitudes in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. Kongelige Danske Videnskabernes Selskab, Biologiske Skrifter, 1948, 5(4):1－34.

[26] Gavin D G, Oswald W W, Wahl E R, et al. A statistical approach to evaluating distance metrics and

analog assignments for pollen records. Quaternary Research, 2003, 60(3): 356－367.

[27] Duda R O, Hart P E, Stork D G. Pattern classification. The 2nd Edition. New York: Wiley - Interscience, 2000, 654.

[28] Cha S H. Comprehensive survey on distance/similarity measures between probability density functions. International Journal of Mathematical Models and Methods in Applied Sciences, 2007, 1(4): 300－307.

附 录

本文使用L-BFGS方法对目标函数$T(\theta)$进行求解，对应当前迭代次的特征－标签矩阵的二阶泰勒展开为：

$$T\left(\theta^{(l+1)}\right)\approx T\left(\theta^{(l)}\right)+\nabla T\left(\theta^{(l)}\right)\Delta+\frac{1}{2}\Delta^{T}H\left(\theta^{(l)}\right)\Delta \tag{9}$$

其中，$\Delta=\theta^{(l+1)}-\theta^{(l)}$，$\nabla T\left(\theta^{(l)}\right)$是梯度矩阵，$H\left(\theta^{(l)}\right)$是Hessian矩阵.

上述二阶泰勒展开的最小化可得：

$$\Delta^{(l)}=-H^{-1}\left(\theta^{(l)}\right)\nabla T\left(\theta^{(l)}\right) \tag{10}$$

在L-BFGS方法中，$\Delta^{(l)}$被视作搜索方向，表示为$\Delta^{(l)}=p^{(l)}$. 设置步长和方向，保证函数值稳定地下降，则有：

$$\theta^{(l+1)}=\theta^{(l)}+\alpha^{(l)}p^{(l)} \tag{11}$$

其中，$p^{(l)}$是搜索方向，$\alpha^{(l)}$是搜索步长. 搜索步长由Wolf condition确定※，式(12)保证找到的步长使目标函数充分减小，式(13)保证斜率充分降低.

$$T\left(\theta^{(l)}+\alpha^{(l)}p^{(l)}\right)\leqslant T\left(\theta^{(l)}\right)+c_1\alpha^{(l)}\nabla T\left(\theta^{(l)}\right)p^{(l)^{T}} \tag{12}$$

$$p^{(l)^{T}}T\left(\theta^{(l)}+\alpha^{(l)}p^{(l)}\right)\leqslant c_2\nabla T\left(\theta^{(l)}\right)p^{(l)^{T}} \tag{13}$$

L-BFGS方法的核心思想是使用一个近似矩阵B用于拟合Hessian矩阵H，以避免复杂的计算，拟合矩阵B如式(14)所示：

$$B^{(l+1)}=\left(I-\rho^{(l)}s^{(l)}\left(u^{(l)}\right)^{T}\right)B^{(l)}\left(I-\rho^{(l)}u^{(l)}\left(s^{(l)}\right)^{T}\right)+\rho^{(l)}s^{(l)}\left(s^{(l)}\right)^{T} \tag{14}$$

其中，I是单位矩阵，其余变量如下所示：

$$\rho^{(l)}=\frac{1}{\left(u^{(l)}\right)^{T}s^{(l)}} \tag{15}$$

$$u^{(l)}=\nabla T\left(\theta^{(l+1)}\right)-\nabla T\left(\theta^{(l)}\right) \tag{16}$$

$$s^{(l)}=\theta^{(l+1)}-\theta^{(l)} \tag{17}$$

（责任编辑　杨可盛）

※Nocedal J, Wright S J. Numerical optimization. The 2nd Edition. New York: Springer Science, 2006, 664.