DOI:10.11992/tis.201706033
网络出版地址:http://kns.cnki.net/kcms/detail/23.1538.TP.20170831.1058.018.html

基于 Spark 的多标签超网络集成学习

李航[1],王进[2],赵蕊[2]
(1.重庆邮电大学 软件工程学院,重庆 400065; 2. 重庆邮电大学 计算智能重庆市重点实验室,重庆 400065)

摘 要:近年来,多标签学习在图像识别和文本分类等多个领域得到了广泛关注,具有越来越重要的潜在应用价值。尽管多标签学习的发展日新月异,但仍然存在两个主要挑战,即如何利用标签间的相关性以及如何处理大规模的多标签数据。针对上述问题,基于 MLHN 算法,提出一种能有效利用标签相关性且能处理大数据集的基于 Spark 的多标签超网络集成算法 SEI-MLHN。该算法首先引入代价敏感,使其适应不平衡数据集。其次,改良了超网络演化学习过程,并优化了损失函数,降低了算法时间复杂度。最后,进行了选择性集成,使其适应大规模数据集。在 11 个不同规模的数据集上进行实验,结果表明,该算法具有较好的分类性能,较低的时间复杂度且具备良好的处理大规模数据集的能力。

关键词:多标签学习;超网络;标签相关性;Apache Spark;选择性集成学习
中图分类号:TP181 **文献标志码**:A **文章编号**:1673-4785(2017)05-0624-16

中文引用格式:李航,王进,赵蕊.基于 Spark 的多标签超网络集成学习[J]. 智能系统学报, 2017, 12(5): 624-639.
英文引用格式: LI Hang, WANG Jin, ZHAO Rui. Multi-label hypernetwork ensemble learning based on Spark [J]. CAAI transactions on intelligent systems, 2017, 12(5): 624-639.

Multi-label hypernetwork ensemble learning based on Spark

LI Hang[1], WANG Jin[2], ZHAO Rui[2]
(1.College of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; 2. Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: Multi-label learning has attracted a great deal of attention in recent years and has a wide range of potential real-world applications, including image identification and text categorization. Although great effort has been expended in the development of multi-label learning, two main challenges remain, i.e., how to utilize the correlation between labels and how to tackle large-scale multi-label data. To solve these challenges, based on the multi-label hypernetwork (MLHN) algorithm, in this paper, we propose a Spark-based multi-label hypernetwork ensemble algorithm (SEI-MLHN) that effectively utilizes label correlation and can deal with large-scale multi-label datasets. First, the algorithm introduces cost sensitivity to enable it to adapt to unbalanced datasets. Secondly, it improves the hypernetwork evolution learning process, optimizes the loss function, and reduces the inherent time complexity. Lastly, it uses selective ensemble learning to enable it to adapt to large-scale datasets. We conducted experiments on 11 datasets wit different scales. The results show that the proposed algorithm demonstrates excellent categorization performance, low time complexity, and the capability to handle large-scale datasets.
Keywords: multi-label learning; hypernetwork; label correlations; Apache Spark; selective ensemble learning

多标签学习在文本分类[1-2]、图像注释[3-4]和生物信息学[5]等多个应用领域得到了广泛关注,也具有越来越重要的应用价值。在多标签学习中,训练集的一个样本均对应一组标签集合。假设 X 表示样本空间,$Y=\{1,2,\cdots,q\}$ 表示所有可能的标签集合,其中标签的总数为 q,$T=\{(x_1,Y_1),(x_2,Y_2),\cdots,(x_m,Y_m)\}$ 为具有 m 个样本的训练集,其中 $x_i\in X$ 且 $Y_i\subseteq Y$。则多标签分类的目标是输出一个多标签分

收稿日期:2017-06-09. 网络出版日期:2017-08-31.
基金项目:重庆市基础与前沿研究计划项目(cstc2014jcyjA40001, cstc2014jcyjA40022);重庆教委科学技术研究项目(自然科学类)(KJ1400436).
通信作者:李航.E-mail:1326202954@qq.com.

类器 $h:x\rightarrow 2^{y}$,使得对每一个给定的实例 $x\in X$,都能预测出合适的标签集合 $Y^{*}\subseteq Y$。

多标签学习的关键挑战在于分类器预测的标签空间数量为指数级(2^{q})。为了解决这个问题,有效地利用不同标签之间的相关性以促进学习过程已成为多标签学习的关键[6-7]。在过去几年,许多利用标签相关性的算法被提出,如校准标签排序(CLR)[8],随机 k 标签集(RAkEL)[9]和广义 k 标签集成(GLE)[10]均考虑了标签之间的相关性,然而这些算法的计算复杂度随标签数量的增加而显著增加。

同时,大部分现有的多标签学习方法没有充分考虑多标签数据的固有属性,即标签类别不平衡。对每一个标签 $y_j\in Y$,令 $D_j^{+}=\{(x_i,+1)\mid y_j\in Y_i,1\leqslant i\leqslant N\}$ 以及 $D_j^{-}=\{(x_i,-1)\mid y_j\notin Y_i,1\leqslant i\leqslant N\}$ 作为正样本和负样本。一般来说,每个类别的正训练样本数远远低于其负训练样本数,这可能导致大多数多标签学习算法的性能降低[11]。文献[12-14]指出,不平衡问题普遍存在于多标签应用中,会损害分类性能。算法交叉耦合聚合学习方法(cross-coupling aggregation, COCOA)[14]同时考虑了标签相关性和不平衡问题,同样其算法复杂度很高,因此如何有效和高效地利用标签间的相关性并削弱不平衡问题的影响仍然是一个悬而未决的问题。

另一方面,目前现实应用中的多标签数据集的样本、特征和标签的数量远远超过常规大小,例如,视频共享网站 Youtube 中有数百万个视频,而每个视频可以被数百万个候选类别标记。然而,大多数多标签学习算法不能很好地适应数据集规模很大的应用。对近 3 年出现的多标签学习方法[15-23]使用的训练集的规模进行统计,可以看出训练样本数在50 000~100 000之间的数据集仅有 5 个,样本数大于100 000的数据集仅有 1 个,大多数现有的多标签学习算法仅适用于处理中小规模数据集。其次,文献[19]虽然利用大规模数据集进行了实验,但是它的计算复杂度高。

多标签超网络 MLHN 与协同演化多标签超网络 Co-MLHN[24]可以挖掘标签间的高阶关系,它将传统的超网络转为多标签超网络,用超边和超边的权重来表示特征子集与标签之间的高阶关系,利用了任意标签间的相关性,且计算复杂度随标签数量的增加呈线性增长,但是其算法时间复杂度与样本数量呈平方级关系,不能很好地处理规模较大的数据集,同时算法也未考虑到标签不平衡对性能的影响。

针对目前多标签超网络存在的问题,本文基于 MLHN 的思想,提出了 Spark 平台下的改进多标签超网络集成算法 SEI-MLHN,有效且高效地解决了多标签学习问题。首先对多标签数据集进行划分;然后对划分后的数据分别用基于 Spark 平台的改进超网络算法 SI-MLHN 进行训练,形成多个局部超网络;最后对多个局部超网络进行选择性集成完成对测试样本的预测。其中,SI-MLHN 利用 MLHN 的思想并在 Spark 平台下进行改进,首先计算每个样本的 k 近邻,然后利用 k 近邻对超网络进行演化学习,得到演化超网络。

为了评估本文算法的性能以及对大规模数据集的适应性,选用不同规模数据集来进行对比实验,验证了本文算法具有良好性能以及具备处理大规模数据集的能力。本文的主要贡献如下:

1)引入了代价敏感,使其能良好地适应多标签不平衡数据,提升算法性能;

2)改良了超网络演化学习过程,大幅度降低 MLHN 算法的计算复杂度;

3)利用选择性集成,降低了时间复杂度,并提高分类性能;

4)基于 Spark 计算框架实现算法,使算法实现并行,提高算法运行效率。

1　相关工作

虽然多标签学习已经成功应用于生物信息学、音频分类[25]以及 web 挖掘[26]等多个领域,但是由于多标签分类器的输出空间为指数级,以及现在大部分应用的数据集规模日益增加,对多标签学习造成了很大的挑战。

为了应对分类器输出空间数量巨大这个问题,现有的方法是利用标签相关性来促进学习过程。基于标签关联性,张敏灵和周志华[27-28]将现有的学习算法分为 3 类,分别为一阶策略、二阶策略以及高阶策略。一阶策略是简单地将多标签学习转为多个独立的二分类问题来解决多标签学习问题,例如 ML-KNN[29]、BR[30]等;二阶策略通过利用标签之间的成对关系解决多标签学习问题,例如 CLR[31]、BP-MLL[32]等;高阶策略通过探索标签之间的高阶关系来解决多标签学习问题,例如 CC[33]、CNMF[34]等。对这 3 种策略进行比较分析,一阶策略的效率

高且概念易理解,但忽略了标签相关性。二阶策略在一定程度上解决了标签相关性,但忽略了现实世界中相关性超过二阶的情况。高阶策略具有比一阶和二阶更强的建模能力,但是其计算复杂度更高,可扩展性更低。

为了应对多标签数据的不平衡性造成算法性能下降这个问题,常规解决方案是为每一个标签训练一个二分类器,并通过随机或合成欠采样/过采样来处理这个二分类器[35-36],但这些方法没有很好地利用标签间的关联性。也有其他的解决方案,如张敏灵等[14]提出交叉耦合聚合算法 COCOA,但是这种算法时间复杂度高,不适合处理大规模数据集。

为了应对数据集规模大这个问题,现有的解决方案是利用分布式存储系统,提供一个基础架构,从而实现高效和可扩展的大数据挖掘与分析。目前,为大数据分析开发了大量的计算框架[37-41],其中,最经典的是 MapReduce[37]。MapReduce 简单、通用且成熟,被广泛使用,但是它只能进行 Map 和 Reduce 计算,不适合描述复杂数据处理过程,数据需要写到磁盘,不能有效地执行迭代算法。为了克服 MapReduce 的缺点,大量的计算框架被设计出来,如 Haloop [38]、Apache Mahout [39]、i2MapReduce [40]和 Apache Spark[41] 等。Haloop 是 Hadoop MapReduce 框架的修改版本,它继承了 Hadoop 的基本分布式计算模型和架构。Apache Mahout 是一个开源项目,主要用于创建可扩展的机器学习算法。i2MapReduce 是 MapReduce 的一个增量处理扩展,并广泛用于大数据挖掘。Apache Spark 是一个开源的集群计算框架,用于大规模的交互计算。在上述框架中,Apache Spark 利用内存计算,并保留 MapReduce 的可扩展性和容错能力,对迭代算法非常有效。Spark 执行速度比 Hadoop MapReduce 快 100 倍[41],并且显著快于其他计算框架。

综上所述,为了解决上述问题,本文使用 Spark 计算框架作为平台来实现多标签算法。

2 Spark 下改进多标签超网络集成算法

MLHN 可以高效地挖掘标签间的关联性且学习复杂度与标签维度呈线性关系,因此本文基于 MLHN 算法提出了 Spark 平台下的改进多标签超网络集成算法 SEI-MLHN,高效地解决了多标签问题。首先,对多标签数据集进行划分,然后对划分后的数据分别用 SI-MLHN 算法进行训练,形成多个局部超网络,最后对多个局部超网络进行选择性集成完成对测试样本的预测。其中,算法 SI-MLHN 利用 MLHN 的思想,在 Spark 平台下进行改进,首先计算每个样本的 k 近邻,然后利用 k 近邻对超网络进行演化学习,得到超网络。本节中,将对算法 MLHN,MLHN 的改进算法 SI-MLHN,以及以 SI-MLHN 为基学习器进行选择性集成的算法 SEI-MLHN 依次进行介绍。

2.1 多标签演化超网络(MLHN)

多标签演化超网络利用超边集合以及超边权重来表示样本特征子集与多标签类别之间的高阶关联。通过演化学习,可以近似地表示训练样本 X 和其标签 Y 之间的概率分布 $P(X,Y)$,在 MLHN 中可以按式(1)进行表示:

$$P(y_i=1\mid\boldsymbol{x})=\frac{P(\boldsymbol{x},y_i=1)}{P(\boldsymbol{x})}=\frac{\sum_{j=1}^{|E|}w_{ji}I(\boldsymbol{x},y_i=1;e_j)}{\sum_{j=1}^{|E|}w_{ji}I(\boldsymbol{x},y_i=1;e_j)+\sum_{j=1}^{|E|}w_{ji}I(\boldsymbol{x},y_i=0;e_j)} \tag{1}$$

式中:y_i 为样本 $\boldsymbol{x}$ 的第 i 个标签;w_{ji} 为超边集合 $|E|$ 中 e_j 的第 i 个权重向量的值;$I(\boldsymbol{x},y_i;e_j)$ 为超边与样本匹配函数,若匹配则取值为 1,反之则为 0,如式(2)所示:

$$I(\boldsymbol{x}_n,y_{ni};e_j)=\begin{cases}1, & \mathrm{dis}(\boldsymbol{x}_n;e_j)\leqslant\delta \text{ 且 } y_{ji}=y_{ni}\\ 0, & \text{其他}\end{cases} \tag{2}$$

式中:y_{ji}是超边 e_j 的第 i 个标签,$\mathrm{dis}(\boldsymbol{x}_n;e_j)$ 为超边 e_j 与样本 $\boldsymbol{x}$ 的欧氏距离,δ 为匹配阈值。δ 的计算方法如式(3):

$$\delta=\frac{\dim(e_j)}{|G_x|\times\dim(\boldsymbol{x})}\sum_{\boldsymbol{x}'\in G_x}\|\boldsymbol{x}-\boldsymbol{x}'\| \tag{3}$$

式中:其中 G_x 为 $\boldsymbol{x}$ 的近邻样本集合,$\dim(\boldsymbol{x})$为样本 $\boldsymbol{x}$ 的特征维度。

为了对未知样本进行预测,MLHN 通常把标签预测误差和相关标签不一致性最小化作为演化学习目标。通过超边初始化、超边替代和梯度下降演化学习来对训练集进行学习,使超边权重 w_{ji}进行更新,流程如图 1 所示。图 1 中,超边 $e_h=(\boldsymbol{v}_h,\boldsymbol{y}_h,\boldsymbol{w}_h)$,$\boldsymbol{v}_h$ 是超边的顶点,为 $\boldsymbol{x}$ 的部分特征;$\boldsymbol{y}_h$ 为 $\boldsymbol{x}$ 的标签;$\boldsymbol{w}_h$ 是 $\boldsymbol{x}_h$ 对应 $\boldsymbol{y}_h$ 的权重向量。

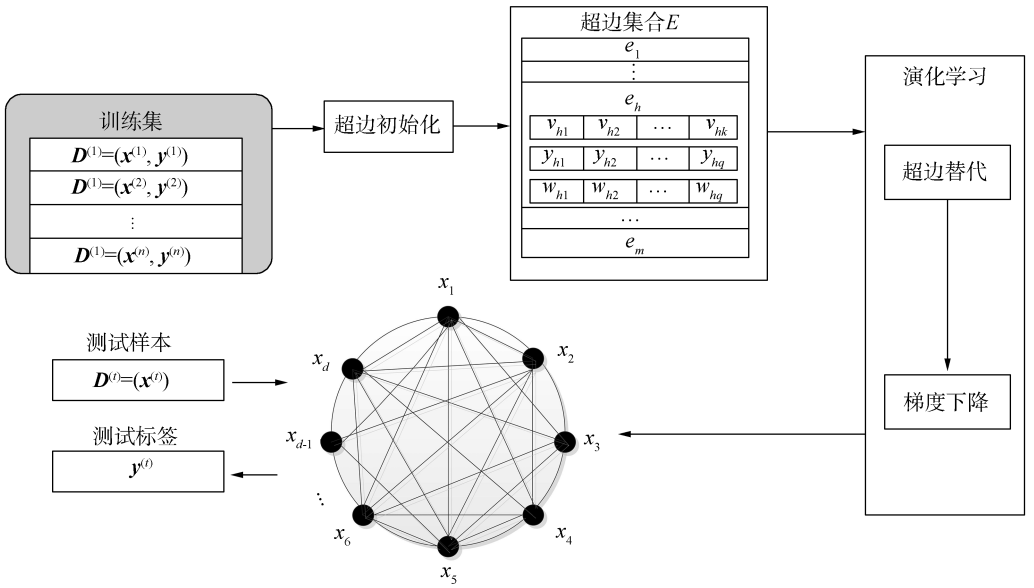


图 1　MLHN 算法流程图

Fig.1　Basic flow chart of MLHN

2.2　Spark 下改进多标签超网络(SI-MLHN)

MLHN 是一种有效的多标签学习算法,但是目前的 MLHN 算法计算复杂度高,且对多标签数据的不平衡特性没有关注。本文一方面改进了 MLHN 的训练过程,引入了代价敏感;另一方面通过并行计算来降低运算时间,设计了 Spark 下改进多标签超网络,记作 SI-MLHN。在本小节,将分别介绍 SI-MLHN 的多标签分类学习过程和演化学习过程。

2.2.1　SI-MLHN 分类学习过程

SI-MLHN 算法关注了多标签样本中普遍存在的标签类别不平衡现象。对于一个未知样本 $\boldsymbol{x}_n$,SI-MLHN将返回每个标签的概率 $P(y_{ni}=1\,|\boldsymbol{x}_n)$,如式(4):

$$P(y_{ni}=1\,|\boldsymbol{x}_n)=\frac{1}{1+\mathrm{e}^{-(W_{ni}^1-W_{ni}^0)}} \tag{4}$$

式中:W_{ni}^1为将样本 $\boldsymbol{x}_n$ 的标签 i 分类为 1 的权重和,W_{ni}^0 则为分类为 0 的权重和,计算方法如式(5)、(6):

$$\boldsymbol{W}_{ni}^1=\sum_{j=1}^{|E|}w_{ji}I(\boldsymbol{x}_n,y_{ni}=1;e_j)\times\cos t_i \tag{5}$$

$$\boldsymbol{W}_{ni}^0=\sum_{j=1}^{|E|}w_{ji}I(\boldsymbol{x}_n,y_{ni}=0;e_j) \tag{6}$$

式中:y_{ni} 为样本 $\boldsymbol{x}$ 的第 i 个标签;w_{ji} 为超边集合 $|E|$中 e_j 的第 i 个权重向量的值;$I(\boldsymbol{x}_n,y_{ni};e_j)$的计算方法如式(2);$\cos t_i$ 为第 i 个标签的代价值,计算方式如式(7):

$$\cos t_i=1+\lg\sum_{(x_n,y_n)\in T}\frac{|\{(x_n,y_n)\,|y_{ni}=0\}|}{|\{(x_n,y_n)\,|y_{ni}=1\}|} \tag{7}$$

式中 T 为有 m 个样本的训练集。

由于 SI-MLHN 采用 sigmoid 函数返回了每个标签与样本相关的概率$P(y_{ni}=1\,|\boldsymbol{x}_n)$,故将相关标签阈值 t_i 设定为 0.5,从而获得每个样本的标签集合,如式(8):

$$y_{ni}^*=\begin{cases}1, & P(y_{ni}=1\,|\boldsymbol{x}_n)\geqslant t_i\\ 0, & \text{其他}\end{cases} \tag{8}$$

在多标签学习中,一个样本只包含标签空间中的部分标签。如果可以排除一些不可能的标签,可以减少标签预测的不确定性。因此,SI-MLHN 借鉴了 Co-MLHN 算法的思想,将 KNN 引入算法,减少算法预测的不确定标签,提高算法的性能。算法 1 为 SI-MLHN 分类学习过程的伪代码。

算法 1　SI-MLHN 分类学习过程

输入　训练集 T,测试样本 x_n,标签数 q,近邻数量 k, SI-MLHN 模型 H,标签阈值 t_i;

输出　标签概率 p,预测标签 y^*。

1)在训练集 T 中计算 $\boldsymbol{x}_n$ 的 k 近邻

2)将模型 H 中是 $\boldsymbol{x}_n$ 的近邻且与 $\boldsymbol{x}_n$ 匹配的超边加入集合 U 中

3)从 U 中提取标签 $y_i=1$ 的超边到集合 U^1 中

4)for $i=1$ to q do

5)$W_1[\mathrm{i}]\leftarrow 0$

6)for each $e_j\in U^1$

7)$W_1[i]=W_1[i]+w_{ji}\times\cos t_i$

8)end for

9)end for

10)从 U 中提取标签 $y_i=0$ 的超边到集 U^0 中

11) for $i=1$ to q do

12) $W_0[\mathrm{i}]\leftarrow 0$

13) for each $e_j \in U^0$

14) $W_0[i]=W_0[i]+w_{ji}$

15) end for

16) end for

17) for $i=1$ to q do

18) $P(y_i=1\,|\boldsymbol{x}_n)=\dfrac{1}{1+\mathrm{e}^{-(W_1[i]-W_0[i])}}$

19) $p[i]=P(y_i=1\,|\boldsymbol{x}_n)$

20) if $P(y_{ni}=1\,|\boldsymbol{x}_n)\geqslant t_i$

21) $\boldsymbol{y}^*[i]=1$

22) else $\boldsymbol{y}^*[i]=0$

23) end if

24) end for

25) return $p,\boldsymbol{y}^*$

2.2.2 SI-MLHN 演化学习过程

SI-MLHN 利用超边的顶点和权重向量来代表多标签数据标签间的高阶关联,其权重向量由超边从训练集中演化学习而来,首先进行了超边初始化,然后进行了超边替代与梯度下降演化学习,并利用 Spark 进行分布式并行计算,通过多个操作技巧,如 cache、broadst,将变量缓存于内存中,大量减少了网络交换数据量和磁盘 I/O 操作,使算法更高效。

在超边初始化的过程中,利用样本$(\boldsymbol{x},\boldsymbol{y})$生成超边 $e=(\boldsymbol{v},\hat{\boldsymbol{y}},\boldsymbol{w})$,超边的顶点向量 $\boldsymbol{v}$ 随机地从样本 $\boldsymbol{x}$ 特征中产生,标签向量 $\hat{\boldsymbol{y}}$ 为样本的标签 $\boldsymbol{y}$,权重向量初始化为 1,表示为 $\boldsymbol{w}=[w_1\ w_2\cdots\ w_q]$,其中$w_i=1.0(1\leqslant i\leqslant q)$。

由于超边顶点向量 $\boldsymbol{v}$ 为随机的,为了更好地拟合训练样本,需要通过超边替代来选择适应度高的超边。如果新生成的超边适应值高于现有超边,则替换该超边。适应值的计算方法如式(9)所示:

$$\mathrm{fitness}(e_j)=\frac{1}{|G|}\sum_{(\boldsymbol{x}_n,\boldsymbol{y}_n)}\frac{1}{q}\sum_{i=1}^{q}|\{i\,|y_{ni}=y_i'\}| \tag{9}$$

式中:超边 e_j 的近邻样本个数为 k,G 为与超边 e_j 匹配的抽样训练集 TS 样本集合,则将 TS 中样本的数量设置为 10 倍的 k,其中 k 个是超边 e_j 的近邻样本,其余的样本则为训练集样本的随机抽样;q 为标签数量;y_{ni}为样本$(\boldsymbol{x}_n,\boldsymbol{y}_n)$的第 i 个标签;y_i'为超边 e_j 的第 i 个标签。由式(9)可以看出适应值代表了超边标签与匹配样本标签的相似度的平均值,相似度越高,则适应值越高。同时,式(4)也展示出,样本与匹配超边的标签相似度越高,被正确分类的概率越大。

本文将预测误差作为学习目标。损失函数如式(10)所示,$P^*(y_{ni}=1\,|\boldsymbol{x}_n)$为 SI-MLHN 分类器对样本$(\boldsymbol{x}_n,\boldsymbol{y}_n)$的第 i 个标签的预测值,利用梯度下降调整超边的权重,降低损失值。超边权重更新为式(11),并通过式(12)~(14),计算 Δw_{ki},其中 w_{ki}为第 k 条超边第 i 个标签的权重,η 为学习速率。

$$\mathrm{Err}_n(W)=\frac{1}{2}\sum_{i=1}^{q}[P^*(y_{ni}=1\,|\boldsymbol{x}_n)-P(y_{ni}=1\,|\boldsymbol{x}_n)]^2 \tag{10}$$

$$w_{ki}=w_{ki}+\Delta w_{ki} \tag{11}$$

$$\Delta w_{ki}=-\eta\frac{\partial \mathrm{Err}_n(W)}{\partial w_{ki}} \tag{12}$$

$$\frac{\partial \mathrm{Err}_n(W)}{\partial w_{ki}}=(P(y_{ni}=1\,|\boldsymbol{x}_n)-P^*(y_{ni}=1\,|\boldsymbol{x}_n))\frac{\partial P^*(y_{ni}=1\,|\boldsymbol{x}_n)}{\partial w_{ki}} \tag{13}$$

$$\frac{\partial P^*(y_{ni}=1\,|\boldsymbol{x}_n)}{\partial w_{ki}}=\frac{\partial\dfrac{1}{1+\mathrm{e}^{-(W_{ni}^1-W_{ni}^0)}}}{\partial w_{ki}}=\frac{1-\dfrac{1}{1+\mathrm{e}^{-(W_{ni}^1-W_{ni}^0)}}}{1+\mathrm{e}^{-(W_{ni}^1-W_{ni}^0)}} \tag{14}$$

算法 2 为 SI-MLHN 的演化学习流程伪代码。由于本文采用欧氏距离作为距离度量,故需要先进行归一化处理。

算法 2 SI-MLHN 演化学习算法

输入 训练集 $T=\{(\boldsymbol{x}_n,\boldsymbol{y}_n)\}(1\leqslant n\leqslant N)$,标签数 q,每个样本生成的超边数 e, 超边替代迭代次数 t_r,随机梯度下降迭代次数 t_d,样本的近邻数量 k;

输出 SI-MLHN:H。

1) $T_{\mathrm{nor}}=T$.map

2) 每条样本进行归一化

3) end map.cache

4) $T_{\mathrm{nor}}\mathrm{Bro}=\mathrm{broadcast}(T_{\mathrm{nor}})$

5) $T_{\mathrm{kv}}=T_{\mathrm{nor}}$.map

6) 每条样本计算与 $T_{\mathrm{nor}}\mathrm{Bro}$ 中样本的欧式距离

7) end map.map

8) 获取距离最近的 k 个样本

9) end map

10) $T_{\mathrm{kv}}\mathrm{Bro}=\mathrm{broadcast}(T_{\mathrm{kv}})$

11) $H_{\mathrm{ini}}=T_{\mathrm{kv}}$.flatmap

12)每条样本生成 e 条超边

13)end flatmap.cache

14) $H_{\text{re}}^{0} = H_{\text{ini}}$ .map

15)对超边进行 t_r 次替代

16)end map.cache

17)for $t \leftarrow 1$ to t_d do

18) $H_{\text{re}}^{t-1}\text{Bro} = \text{broadcast}(H_{\text{re}}^{t-1})$

19) $H_{\text{tmp}} = T_{\text{kv}}$ .map

20)对样本利用 H_{re}^{t-1}Bro 中超边集合预测

21)end map.flatmap

22)对超边进行梯度计算

23)end flatmap.reduceByKey(合并梯度值)

24) $H_{\text{re}}^{t} = H_{\text{re}}^{t-1}$ .leftjoin(H_{tmp})

25).map

26)利用合并后梯度值更新超边权重

27)end map

28)end for

29) $H = H_{\text{re}}^{t}$

30)return H

T_{nor} 为训练集经过分布式并行归一化处理后的结果;broadcast 操作可以保存只读变量,并保在每个节点内存中; T_{nor}Bro 为归一化后样本广播变量; T_{kv}Bro 为样本与其 k 近邻的键值对 T_{kv} 的广播变量; H_{ini}为超边初始化集合; H_{re}^{0} 为超边替代后集合; H_{re}^{t} 为超边演化学习后集合。

2.3　Spark 下集成多标签超网络(SEI-MLHN)

SI-MLHN 为 MLHN 的 Spark 平台下分布式并行改进方法,大幅缩短了训练时间,但是其时间复杂度仍然随着样本数量的增加呈平方级增长,仍然无法很好适应大样本数据。故本文利用选择性集成,一方面降低时间复杂度,另一方面提高算法性能,提出了 Spark 下集成多标签超网络,记作 SEI-MLHN。SEI-MLHN 首先将训练集进行分簇,并分别用 SI-MLHN 算法演化学习多个局部多标签超网络。对于未知样本,首先获得近邻簇,然后利用局部超网络选择性集成对测试样本进行预测,SEI-MLHN 的流程见图 2。

为了对训练样本进行分簇,本文选择了基于神经网络的无监督聚类方法自组织神经网络(SOM)[42]。SOM 对类簇初始化不敏感,并且可以很好地发现数据之间的结构关系,为了让其适应大规模数据处理将其进行了 Spark 下并行化扩展,记为 S-SOM。算法 3 为算法伪代码,首先利用每个分区计算获胜节点,并更新优胜邻域中的输出单元,然后将各个分区的值利用 reduce 算子进行合并,并更新优胜邻域,达到终止条件得到最终输出层。其中,计算邻域函数选取高斯函数,距离仍然选取欧氏距离, WBro 为 SOM 初始化输出层 W 的广播变量在 6)~12)中完成了一次迭代计算, W_c^t 为每次迭代利用每个分区样本更新后输出层, W_r^t 为分区合并的输出层。

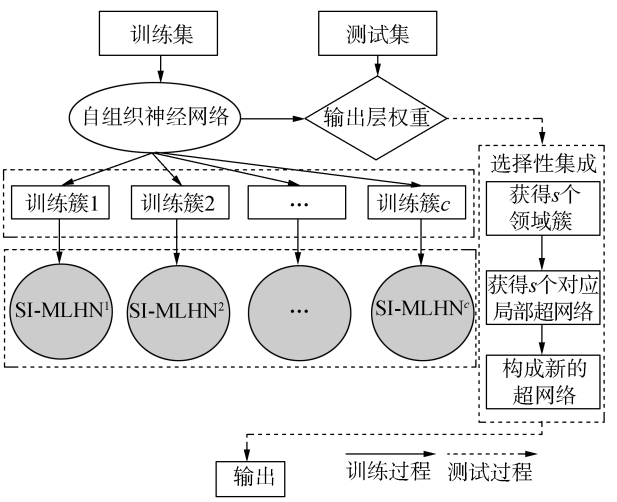


图 2　SEI-MLHN 流程图

Fig.2　Flow chart of SEI-MLHN

算法 3　S-SOM

输入　训练集 $T = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}(1 \leqslant n \leqslant N)$,类簇个数 c ,学习率 η,迭代次数 t_i;

输出　类簇 $T_1', T_2', \cdots, T_s'$。

1) $T_{\text{nor}} = T$ .map

2)每条样本进行归一化

3)end map.cache

4) W =初始化输出层(随机抽取 c 个样本)

5) WBro =broadcast(W)

6)for $t \leftarrow 1$ to t_i

7)for each T_{nor}的分区 do

8)for each 分区内样本 do

9)利用输出层计算获胜节点

10)更新优胜邻域中的值为 W_c^t

11)end for

12)end for

13)更新优胜邻域

14) W_r^t =利用 reduce 合并各分区 W_c^t

15)end for

16) $T_1', T_2', \cdots, T_p'$ = T_{nor},计算获胜节点(W_r^t)

17)return $T_1', T_2', \cdots, T_s'$

完成训练集分簇后,利用算法 2 对簇构建 SI-MLHN 超网络。对于测试集,SEI-MLHN 将进行选择性集成,伪代码见算法 4。

算法 4 SEI-MLHN 分类算法

输入 测试集 $E=\{\boldsymbol{x}_n\}(1\leqslant n\leqslant M)$,样本数量为 M,样本近邻数量 k , SEI-MLHN:$H=\{H_1',H_2',\cdots,H_3'\}$,簇数为 c,邻域簇数为 s,S-SOM 输出为 $T_1',T_2',\cdots,T_c'$;

输出 $E^*=\{(\boldsymbol{x}_n,p_n,\boldsymbol{y}_n^*)\}(1\leqslant n\leqslant M)$,其中 p 为测试集标签概率,$\boldsymbol{y}^*$ 为测试集预测标签。

1)E 在 $T_1',T_2',\cdots,T_c'$中计算 s 个最近邻簇,并将其加入 U^s,簇对应的超网络为 $H_1',H_2',\cdots,H_s'$

2)for each $T_i'\in U^s$ do

3) E_{kv}^i =寻找 E 中样本在 T_i' 在中的 k 近邻

4) $E_{\mathrm{kh}}^i=E_{\mathrm{kv}}^i$ .flatmap

5)组成测试样本与近邻样本对

6)end flatmap.leftjoin(H'_i).reduceByKey

7)end for

8)将 s 个 E_{kh}^i 合并为集合 F

9)$E^*=F$.reduceByKey( )

10).map

11)从 $s*k$ 近邻中选取最近 k,利用算法 1 进行预测

12)end map

13)return E^*

算法 4 中,对测试样本选取了 s 个最近类簇产生的局部超网络并把局部组合起来,然后进行预测,得到分类结果。其中, E_{kv}^i 为测试样本与其在第 i 个簇中的 k 近邻组成的键值对, E_{kh}^i 为测试样本与其在第 i 个簇中的 k 近邻以及超边组成的键值对。

2.4 时间复杂度分析

SEI-MLHN 利用 SI-MLHN 进行选择性集成来提高学习器的稳定性和泛化能力。对于含有 N 个训练样本,样本特征纬度为 d,标签数量为 q 的训练集,SI-MLHN 的训练复杂度为 $O(N^2d+enkN+kqN)$,记为 $F_{SI}(N,k,e,n,d,q)$,其中 e 为每个训练样本产生的超边数量,k 为近邻数量,n 为训练样本的抽样,在大规模数据集中 $n\ll N$ 。对于未知样本进行预测时,SI-MLHN 首先在训练集中寻找 k 个近邻样本,然后利用与之匹配的近邻样本产生的超边进行预测。因此对有 M 个样本的测试集,SI-MLHN 的预测时间复杂度为 $O(MNd+eMN+kqM)$,记为 $F_{\mathrm{SI}}'(M,k,e,N,d,q)$。SEI-MLHN 对训练集进行了分簇,并在预测时删除了冗余学习器,因此其训练时间复杂度为 $O(c\cdot F_{\mathrm{SI}}(N',k,e,n,d,q))$,测试时间复杂度为 $O(s\cdot F_{\mathrm{SI}}'(M,k,e,N',d,q))$,其中 c 为训练集聚类簇数,s 为邻域簇的数量, N' 为最大类簇中样本的数量, N' 的数量取决于 c 以及训练数据的分布,一般接近于 N/c 。

3 实验

3.1 数据集

为了对算法性能进行全面的评估,本文选择了 11 个公开的常用多标签数据集进行实验,其中训练样本数小于 5 000 的数据集有 6 个,大于 100 000 的数据集有 2 个,如表 1 所示,表中的标签基数是指每个样本关联标签的平均数量。由于文本数据具有高维稀疏的特性,故在表 1 中对所有的文本数据集均使用 Lee 和 Jiang[43] 提出的模糊相关度量进行了变换,在模糊变换之后,每个文档由模糊相关性向量表示,且维度与标签维度相同。

表 1 实验使用的多标签数据集

Table 1 Multilabel data sets used in experiments

数据集	样本数	属性数	标签数	标签基数	领域
emotions	593	72	6	1.869	Music
Scene	2 407	294	6	1.074	Images
Yeast	2 417	103	14	4.237	Biology
Medical	978	1 449	45	1.245	Text
Enron	1 702	1 001	53	3.378	Text
CAL500	502	68	174	26.044	Music
Eurlex-sm	19 348	5 000	201	2.213	Text
Eurlex-dc	19 348	5 000	412	1.292	Text
Mediamil	43 907	120	101	4.376	Video
Nuswide-bow	269 648	500	81	1.869	Images
Nuswide-cVLADplus	269 648	128	81	1.869	Images

3.2 评价指标

假设 X 表示样本空间, $Y=\{1,2,\cdots,q\}$ 表示所有可能的标签集合, $E=\{(\boldsymbol{x}_i,Y_i)\mid 1\leqslant i\leqslant M\}$ 为具有 M 个样本的多标签测试集, h 为输出的多标签分类器,则测试样本 $\boldsymbol{x}_i$ 的预测结果为 $h(\boldsymbol{x}_i)$。$f(\boldsymbol{x}_i,y)$ 是标签 y 在样本 $\boldsymbol{x}_i$ 上排名质量的实值函数,例如对

于任意 $y_1 \in Y$ 以及 $y_2 \in Y$ 而言,$f(\boldsymbol{x}_i, y_1) > f(\boldsymbol{x}_i, y_2)$ 成立。实值函数 $f(\cdot,\cdot)$ 也可以转为排序函数 $\text{rank}_f(\cdot,\cdot)$,即将所有的实值输出 $f(\boldsymbol{x}_i, y)$ 映射到集合 Y 上,使得 $f(\boldsymbol{x}_i, y_1) > f(\boldsymbol{x}_i, y_2)$ 时,$\text{rank}_f(\boldsymbol{x}_i, y_1) > \text{rank}_f(\boldsymbol{x}_i, y_2)$ 也成立。基于上述描述,本文采用的多标签性能评价指标如下。

1)Hamming Loss:用于考察样本在单个标签上的误分类情况。

$$\text{hloss}(h) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{Q}|h(x_i)\Delta Y_i|$$

式中 Δ 表示两个集合的对称差。

2)One-error:用于考察样本测标签排序集合中最前端的标签误分类的情况。

$$\text{one-error}_s(f) = \frac{1}{M}\sum_{i=1}^{M}[(\arg\max_{y\in Y} f(x_i, y)) \notin Y_i]$$

3)Ranking Loss:用于考察样本的预测标签排序集合中的错排情况。

$$\text{rloss}_s(f) = \frac{1}{M}\sum_{i=1}^{M}\frac{L}{|Y_i||\overline{Y_i}|}$$

$$L = (y_1, y_2)|f(\boldsymbol{x}_i, y_1) \leqslant f(\boldsymbol{x}_i, y_2), (y_1, y_2) \in Y_i \times \overline{Y_i}$$

式中 $\overline{Y_i}$ 是 Y_i 的补集。

4)Average Precision:用于考察样本的预测标签集合中排在该样本标签之前的标签分类正确的情况。

$$\text{avgprec}_s(f) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{|Y_i|}\cdot \sum_{y\in Y_i}\frac{|\{y'|\text{rank}_f(x_i, y') \leqslant \text{rank}_f(x_i, y), y' \in Y_i\}|}{\text{rank}_f(x_i, y)}$$

5) Example Based F_1:

$$F_1 = \frac{1}{M}\sum_{i=1}^{M}\frac{2|Y_i \cap h(x_i)|}{|Y_i| + |h(x_i)|}$$

3.3　比较的算法和实验环境

在本文中,将 SEI-MLHN 与两种系列的算法进行比较实验,第 1 种系列是常用的多标签学习算法,如表 2 的前 8 种算法所示,它们均在表 3 所示的单节点环境中实现。第二种系列是在 Spark 平台下实现的超网络多标签学习算法,如 S-CoMLHN 以及 SEI-MLHN 的基学习器 SI-MLHN,Spark 集群环境如表 4 所示,其中 S-CoMLHN 是 Co-MLHN 在 Spark 平台下的实现。

在实验过程中比较算法的参数设置取自文献[8-9,14,24,30,39,44],详细设置见表 3,其中算法 SI-MLHN 中每个样本生成超边的数量 e 为 10,超边替代迭代次数 t_r 为 5,随机梯度下降迭代次数 t_d 为 5。

表 2　实验中的比较算法

Table 2　Comparison of algorithms used in experiments

算法	参数设置
BRSVM[30]	基学习器:线性支持向量机
ML-KNN[29]	smooth=1.0, 近邻数=10
CLRSVM[8]	基学习器:线性支持向量机
RAkEL[9]	标签集 k=3, 集成大小= $2q$
ECC[33]	基学习器:线性支持向量机 集成大小=50
IBLR[44]	近邻数=10
COCOA[14]	k=10
S-CoMLHN[24]	近邻数=20,η=0.01
SI-MLHN	近邻数=10, η=0.01

表 3　实验单节点环境

Table 3　Experimental environment

处理器型号	单 CPU 核心数	CPU 频率/GHz	CPU 个数	内存容量/GB	是否支持超线程
E5-2620	6	2.0	2	64	是

表 4　集群环境配置

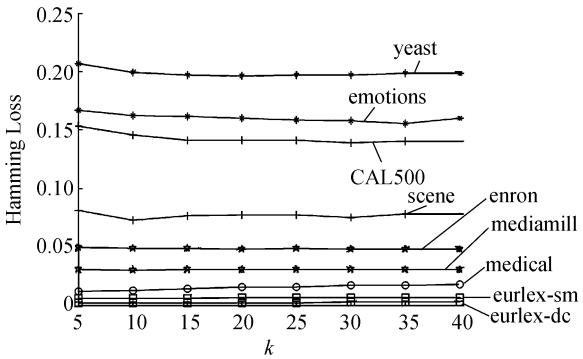
Table 4　Configuration in clustered environment

集群配置	版本号
操作系统	Cent OS6.5
节点个数	16
Scala 版本	2.10.4
Hadoop 版本	2.5.2
Spark 版本	1.5.1
JDK 版本	1.7

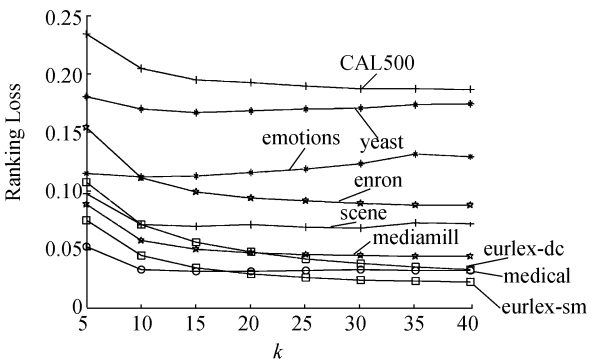
3.4　实验结果

3.4.1　参数分析

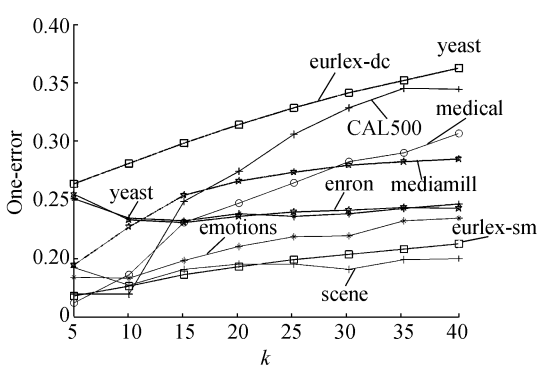
对于算法 SEI-MLHN,近邻个数 k 是关键参数之一,为了测试算法对 k 参数的敏感度,本节将 k 以 5 为步长,在 5~40 范围内测试算法 SI-MLHN 的性能。如图 3 所示,(c)、(e)随 k 的增大波动较大,部分数据集 k>10 后趋于稳定,多数数据集随 k 增大性能变差;(a)中对 k 值不敏感,几乎没有变化;(b)随 k 值增大性能变好,且在 k > 10 后趋于平稳;在(d)中不同的数据集随着 k 值的变化,性能既有增加的,也有降低的,但总体上在 k=10 时,能有较好的性能。这是由于 k 值直接关系着预测样本的匹配超边数量,当 k 太小时与之匹配的超边数量太小,会漏掉一些相关标签,k 值太大则会引入噪声标签且会影响算法运行效率,故本文将 k 取为 10。



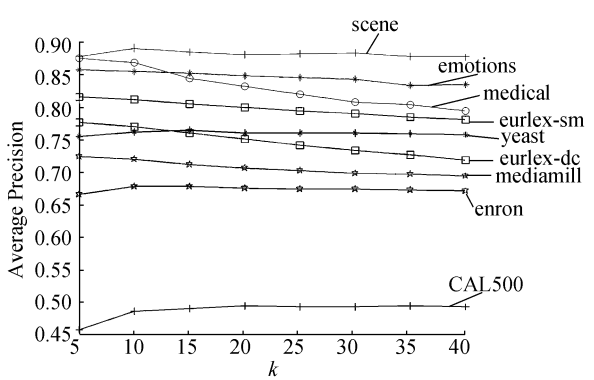
(a)算法 SEI-MLHN 在各个数据集下对应不同 k 值的 Hamming Loss 变化情况



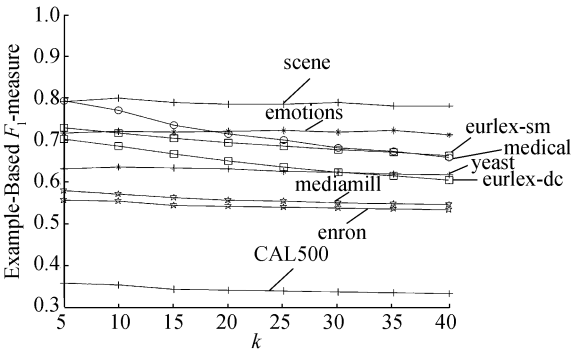
(b)算法 SEI-MLHN 在各个数据集下对应不同 k 值的 Ranking Loss 变化情况



(c)算法 SEI-MLHN 在各个数据集下对应不同 k 值的 One-error 变化情况



(d)算法 SEI-MLHN 在各个数据集下对应不同 k 值的 Average Precision 变化情况

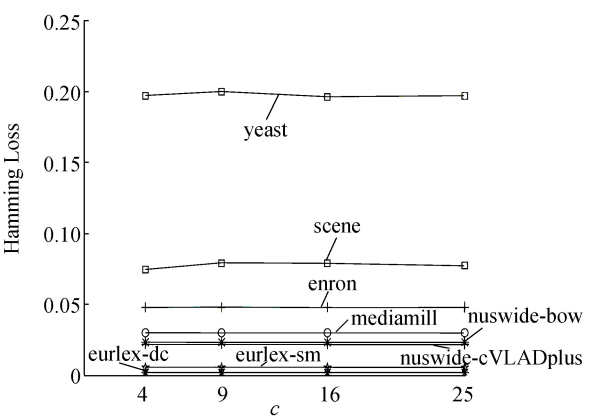


(e)算法 SEI-MLHN 在各个数据集下对应不同 k 值的 Example Based F_1 变化情况

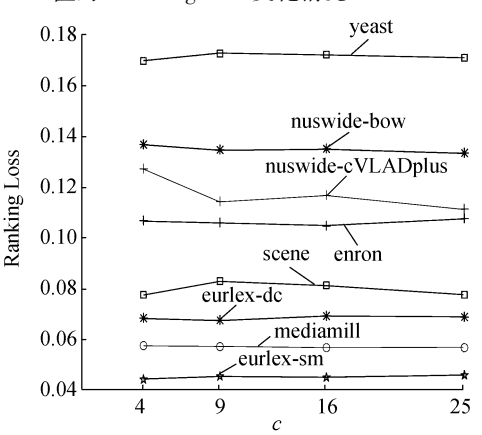
图 3 算法 SEI-MLHN 在各个数据集下对应不同 k 值的分类性能比较

Fig.3 Performance comparison of SEI-MLHN under different values of k on Data Sets

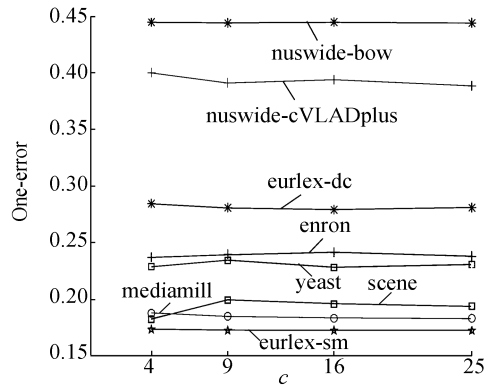
对于 SEI-MLHN 簇数 c,由于本文采用自组织神经网络进行分簇,将训练数据映射到二维空间,故本文将 c 设置为完全平方数,且将选择分类器个数 s 设为$\sqrt{c}$。为了测试算法对分簇数 c 与基分类器个数 s 的敏感度,将 c 设为 2~25 的完全平方数,实验结果如图 4。



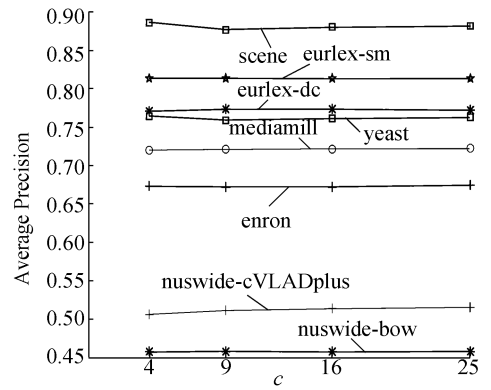
(a)算法 SEI-MLHN 在各个数据集下对应不同 c 值的 Hamming Loss 变化情况



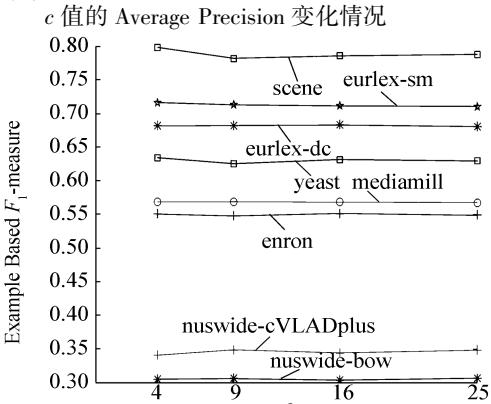
(b)算法 SEI-MLHN 在各个数据集下对应不同 c 值的 Ranking Loss 变化情况



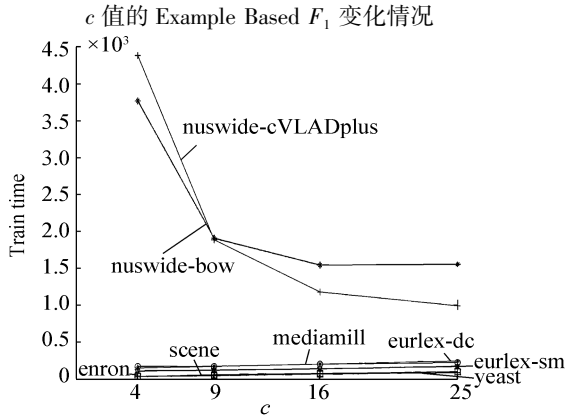
(c)算法 SEI-MLHN 在各个数据集下对应不同 c 值的 One-error 变化情况



(d)算法 SEI-MLHN 在各个数据集下对应不同 c 值的 Average Precision 变化情况



(e)算法 SEI-MLHN 在各个数据集下对应不同 c 值的 Example Based F_1 变化情况



(f)算法 SEI-MLHN 在各个数据集下对应不同 c 值的 Train time 变化情况

图 4　算法 SEI-MLHN 在各个数据集下对应不同 c 值的分类性能比较

Fig.4　Performance comparison of SEI-MLHN under different values of c on Data Sets

图 4 中,(a)、(c)、(d)、(e)中算法 SEI-MLHN 在各个数据集中的性能几乎不变或变化幅度非常小,说明 Hamming Loss 、One-error、Average Precision 和 Example Based F_1 对 c 和 s 的变化不敏感,(b)中 SEI-MLHN 的性能在数据集 nuswide-cVLADplus 中随 c 和 s 的增大有小幅度的优化,(f)中可以看出较小规模数据随簇数 c 和 s 的增加训练时间小幅增加,而对于大规模数据集,算法训练时间大幅减少。故本文对样本数量小于 10 000 的训练集,c 和 s 分别选取 4 和 2;样本数量在 10 000 到 100 000 之间的训练集,c 和 s 分别选取 9 和 3;样本数量大于 100 000的训练集,c 和 s 分别选取 25 和 5。

3.4.2　分类性能比较

在实验中,所有的多标签学习算法均采用相同的数据划分,用 50%的数据进行训练,其余 50%的数据进行测试。在计算评价指标时,评价指标取重复 10 次实验的平均值。由于比较的部分算法无法在一周内对数据集 nuswide-cVLADplus、nuswide-bow 完成训练预测,故在对算法性能进行比较与分析时只取其余 9 个数据集进行实验,如表 5 所示。在表 6 中列出了本文算法在 nuswide-bow、nuswide-cVLADplus 数据集上的评价指标值。

表 5 为 5 个不同评价指标下各个多标签学习算法在常规规模的数据集上的学习性能,表 5 中评价指标后的“↓”表示指标取值越小性能越佳,符号“↑”表示指标取值越大性能越佳,其中 AveR 表示该算法的平均排名。此外,表 7 对每个学习算法进行编号,例如 BRSVM(A_1)表示用 A_1 代表算法 BRSVM,再进一步给出了各多标签学习算法之间的相对性能,具体为,给定算法 A_1 和 A_2,$A_1>A_2$ 表示在给定的评价指标上,基于显著度 0.05 的威尔科克森符号秩检验(Wilcoxon signed rank test),算法 A_1 的性能显著优于 A_2。本文通过打分的方式对各学习算法的性能进行总体评价,若 $A_1 > A_2$,则 A_1 的分数加 1,A_2 的分数减 1,通过比较每个算法的最终分数,可以对算法进行排序,其中 A_1 分数高于 A_2,表示算法 A_1 的总体性能优于 A_2。

从表 7 中可以发现,算法 SEI-MLHN 在除 Ranking loss 外的各项指标以及总分均显著高于其余算法,表明它的分类性能在总体上优于其余算法。其次,算法 SI-MLHN 在 Example based F_1 上明

显优于 S-CoMLHN,说明 SI-MLHN 在一定程度上削弱了标签不平衡的影响,虽然算法 SI-MLHN 在 Hanmming Loss 和 Ranking loss 上稍劣于 S-CoMLHN,但从整体性能上进行比较,算法 SI-MLHN 仍优于 S-CoMLHN。再次,算法 COCOA 考虑了标签间的高阶关系且削弱了不平衡性的影响,但在 Hanmming Loss 上的性能较差,导致总分稍低于 S-CoMLHN,也说明其总体性能稍低于 S-CoMLHN。最后,可以从表 7 中得出结论,本文所提出的算法 SEI-MLHN 具有很好的处理多标签分类问题的能力。

3.4.3 运行效率比较

为了对算法 S-CoMLHN、SI-MLHN 以及 SEI-MLHN 的运行时间进行比较,本文用 3 个较大规模数据集(eurlex-sm、eurlex-dc、mediamill)和 2 个大规模数据集(nuswide-bow 、nuswide-cVLADplus)进行实验,且令所有实验的 SparkTask 为 32。由于 S-CoMLHN 在 24 小时内无法完成 2 个大规模数据集的训练和预测,故图 5 中无对比结果。

从图 5(a)中观察到,算法 S-CoMLHN 的训练时间显著高于 SI-MLHN 和 SEI-MLHN,且在数据集 nuswide-bow 和 nuswide-cVLADplus 上没有得到有效结果,说明该算法不能很好地适应大规模数据。在数据集 eurlex-sm、eurlex-dc-leaves、mediamill 上,算法 SEI-MLHN 与 SI-MLHN 的训练时间没有较大的区别,但是当数据集规模增大时,算法 SEI-MLHN 的训练时间明显低于 SI-MLHN,表明算法 SEI-MLHN 比 SI-MLHN 大大缩短了训练时间。从图 5(b)中观察到,在数据集 eurlex-sm、eurlex-dc、mediamill 上,S-CoMLHN、SI-MLHN 以及 SEI-MLHN 的测试时间没有较大的区别,但是当数据集规模增大后,S-CoMLHN 无法得到有效结果,且算法 SEI-MLHN 的测试时间也明显低于 SI-MLHN,说明算法 SEI-MLHN 也缩短了算法的测试时间。从图 5 中可以得到结论,算法 SEI-MLHN 具有较高的算法运行效率,且对大规模数据集具有良好的适应能力。

由于不同的 SparkTask 数值会对算法的运行时间产生影响,因此,为了测试 SparkTask 对算法 SEI-MLHN 运行效率的影响,选择了不同大小规模的数据集进行实验。图 6 给出了随不同的任务数,算法 SEI-MLHN 在各个数据集下的训练与测试时间的变化情况,时间以秒为单位。从图 6 中观察得到,对规模较小的多标签数据集,如 emontions、yeast,不同的 SparkTask 数值对算法的运行时间没有较大的影响。而对于较大规模数据集 mediamil,随着 SparkTask 数值的增加,算法的训练时间与测试时间急速下降,最后趋于平稳,但是当 SparkTask 增加到一定数量时训练测试时间均增大,这是因为当数据集规模不够大,SparkTask 达到某个饱和点时,集群上各个节点的通信时间远远大于计算时间,算法的性能趋于平缓甚至变弱。

表 5 比较算法在 5 个评价指标上的实验结果(均值、排名)

Table 5 Experimental results of each comparing algorithm(mean rank) on five evaluation metrics

算法	Hamming Loss ↓									
	CAL500	emotions	medical	enron	scene	yeast	eurlex-sm	mediamill	eurlex-dc	AveR
BRSVM	0.137(1)	0.205(6)	0.012(2)	0.046(1.5)	0.11(8)	0.202(7.5)	0.006(4.5)	0.031(5)	0.002(6.5)	4.67
CLRSVM	0.141(3.5)	0.206(7)	0.022(10)	0.046(1.5)	0.111(9)	0.201(6)	0.008(9.5)	0.035(10)	0.003(9)	7.28
RAkEL	0.170(8)	0.262(10)	0.014(7)	0.058(10)	0.141(10)	0.254(10)	0.008(9.5)	0.0307(5)	0.002(6.5)	8.44
ECC	0.141(3.5)	0.200(4)	0.011(1)	0.047(3)	0.092(6)	0.202(7.5)	0.005(1)	0.0302(2)	0.002(6.5)	3.83
IBLR	0.232(10)	0.204(5)	0.014(7)	0.057(9)	0.087(4)	0.199(3.5)	0.007(8)	0.032(8)	0.006(10)	7.17
COCOA	0.173(9)	0.216(9)	0.013(4)	0.053(8)	0.094(7)	0.212(9)	0.006(4.5)	0.034(9)	0.002(1)	6.72
MLKNN	0.140(2)	0.209(8)	0.014(7)	0.049(7)	0.091(5)	0.199(3.5)	0.006(4.5)	0.032(7)	0.002(2)	5.11
S-CoMLHN	0.142(5)	0.169(2)	0.017(9)	0.048(4.5)	0.078(2)	0.196(1)	0.006 1(7)	0.030 5(3)	0.001 9(4)	4.17
SI-MLHN	0.147(7)	0.159(1)	0.014(5)	0.049(6)	0.082 (3)	0.200(5)	0.006(4.5)	0.031(5)	0.002(6.5)	4.78
SEI-MLHN	0.146(6)	0.163(3)	0.013(3)	0.048(4.5)	0.075(1)	0.198(2)	0.006(2)	0.030(1)	0.002(3)	2.83

续表 5

算法	Ranking Loss ↓									
	CAL500	emotions	medical	enron	scene	yeast	eurlex-sm	mediamill	eurlex-dc	AveR
BRSVM	0.498(10)	0.286(9)	0.165(10)	0.303(9)	0.179(9)	0.324(9)	0.273(9)	0.229(9.5)	0.288(9)	9.28
CLRSVM	0.207(7)	0.162(4)	0.124(8)	0.075(1)	0.080(5)	0.170(4.5)	0.127(7)	0.105(7)	0.035(3)	5.17
RAkEL	0.472(9)	0.315(10)	0.150(9)	0.306(10)	0.209(10)	0.353(10)	0.342(10)	0.229(9.5)	0.292(10)	9.72
ECC	0.195(3)	0.165(5)	0.059(7)	0.130(8)	0.081(7)	0.182(8)	0.143(8)	0.177(8)	0.175(8)	6.89
IBLR	0.442(8)	0.170(7)	0.058(6)	0.111(6)	0.078(4)	0.171(6)	0.058(6)	0.054(4)	0.126(7)	6
COCOA	0.202(5)	0.168(6)	0.038(3)	0.084(2)	0.073(2)	0.170(4.5)	0.015(1)	0.046(1)	0.021(1)	2.83
MLKNN	0.186(1)	0.181(8)	0.050(5)	0.086(3)	0.082(8)	0.172(7)	0.022(2)	0.056(5)	0.033(2)	4.56
S-CoMLHN	0.191(2)	0.114(3)	0.032(1)	0.095(4)	0.071(1)	0.168(1)	0.030(3)	0.049(2)	0.046(5)	2.44
SI-MLHN	0.202(4)	0.104(1)	0.045(4)	0.113(7)	0.080(6)	0.169(2)	0.048(5)	0.049(3)	0.046(4)	4
SEI-MLHN	0.204(6)	0.112(2)	0.033(2)	0.107(5)	0.078(3)	0.170(3)	0.045(4)	0.057(6)	0.068(6)	4.11
算法	One-error ↓									
	CAL500	emotions	medical	enron	scene	yeast	eurlex-sm	mediamill	eurlex-dc	AveR
BRSVM	0.394(8)	0.336(9)	0.314(8)	0.339(8)	0.357(9)	0.258(9)	0.271(8)	0.203(6.5)	0.408(7)	8.72
CLRSVM	0.233(6)	0.27(4)	0.642(10)	0.212(1)	0.248(8)	0.23(2)	0.447(10)	0.261(10)	0.897(10)	7.17
RAkEL	0.777(9)	0.417(10)	0.319(9)	0.499(10)	0.435(10)	0.272(10)	0.387(9)	0.203(6.5)	0.42(9)	9.39
ECC	0.143(2)	0.272(5)	0.169(1)	0.252(7)	0.24(7)	0.244(8)	0.157(2)	0.17(2)	0.281(3)	5.67
IBLR	0.867(10)	0.293(6)	0.219(6)	0.3985(9)	0.228(5)	0.238(5)	0.257(7)	0.181(4)	0.413(8)	6.61
COCOA	0.15(3)	0.296(7.5)	0.179(2)	0.247(6)	0.224(4)	0.234(3)	0.13(1)	0.157(1)	0.208(1)	3.17
MLKNN	0.118(1)	0.296(7.5)	0.206(4)	0.232(2)	0.239(6)	0.241(7)	0.161(3)	0.179(3)	0.264(2)	5.17
S-CoMLHN	0.287(7)	0.210(3)	0.271(7)	0.235(3)	0.196(2)	0.237(4)	0.195(6)	0.217(8)	0.324(5)	5
SI-MLHN	0.202(5)	0.166(1)	0.215(5)	0.239(5)	0.203(3)	0.241(6)	0.184(5)	0.231(9)	0.325(6)	5
SEI-MLHN	0.170(4)	0.183(2)	0.186(3)	0.237(4)	0.183(1)	0.229(1)	0.173(4)	0.185(5)	0.282(4)	3.11
算法	Average Precision ↑									
	CAL500	emotions	medical	enron	scene	yeast	eurlex-sm	mediamill	eurlex-dc	AveR
BRSVM	0.295(8)	0.725(9)	0.682(9)	0.488(9)	0.765(9)	0.666(9)	0.594(8)	0.501(9.5)	0.557(8)	8.72
CLRSVM	0.453(7)	0.8(4.5)	0.416(10)	0.699(1)	0.855(8)	0.757(6)	0.436(10)	0.530(8)	0.236(10)	7.17
RAkEL	0.267(9)	0.682(10)	0.697(8)	0.442(10)	0.718(10)	0.629(10)	0.499(9)	0.501(9.5)	0.548(9)	9.39
ECC	0.492(2)	0.8(4.5)	0.844(4)	0.665(7)	0.858(6.5)	0.754(8)	0.758(6)	0.601(7)	0.701(6)	5.67
IBLR	0.253(10)	0.79(6)	0.814(7)	0.605(8)	0.864(5)	0.758(4.5)	0.752(7)	0.696(5)	0.643(7)	6.61
COCOA	0.471(6)	0.789(7)	0.863(2)	0.683(2)	0.868(4)	0.758(4.5)	0.848(1)	0.728(1)	0.827(1)	3.17
MLKNN	0.484(4)	0.781(8)	0.833(5)	0.672(6)	0.858(6.5)	0.756(7)	0.815(2)	0.693(6)	0.779(2)	5.17
S-CoMLHN	0.496(1)	0.849(3)	0.815(6)	0.675(5)	0.882(2)	0.762(2)	0.795(5)	0.704(3)	0.741(4)	3.44
SI-MLHN	0.484(5)	0.868(1)	0.850(3)	0.675(4)	0.875(3)	0.761(3)	0.803(4)	0.698(4)	0.740(5)	3.56
SEI-MLHN	0.486(3)	0.856(2)	0.870(1)	0.677(3)	0.887(1)	0.765(1)	0.814(3)	0.722(2)	0.772(3)	2.11

续表 5

算法	Example Based F_1 ↑									
	CAL500	emotions	medical	enron	scene	yeast	eurlex-sm	mediamill	eurlex-dc	AveR
BRSVM	0.336(6)	0.583(9)	0.657(8)	0.534(7)	0.610(9)	0.606(8)	0.642(8)	0.525(7.5)	0.568(7)	7.72
CLRSVM	0.252(10)	0.590(7)	0.263(10)	0.537(5)	0.618(8)	0.610(7)	0.335(10)	0.382(10)	0.087(10)	8.56
RAkEL	0.347(4)	0.525(10)	0.678(7)	0.497(9.5)	0.549(10)	0.542(10)	0.555(9)	0.525(7.5)	0.552(8)	8.33
ECC	0.345(5)	0.635(4)	0.720(5)	0.571(1)	0.692(4)	0.622(3)	0.678(4)	0.508(9)	0.614(3)	4.22
IBLR	0.32(9)	0.595(6)	0.726(3)	0.497(9.5)	0.681(5)	0.612(6)	0.658(7)	0.532(5)	0.573(6)	6.28
COCOA	0.369(1)	0.624(5)	0.724(4)	0.542(4)	0.66(6.5)	0.613(5)	0.721(1)	0.554(3)	0.686(1)	3.39
MLKNN	0.323(8)	0.585(8)	0.651(9)	0.523(8)	0.66(6.5)	0.603(9)	0.673(5)	0.528(6)	0.584(5)	7.17
S-CoMLHN	0.331(7)	0.709(3)	0.688(6)	0.535(6)	0.781(3)	0.619(4)	0.670(6)	0.546(4)	0.525(9)	5.33
SI-MLHN	0.359(2)	0.730(1)	0.752(2)	0.551(2)	0.785(2)	0.635(2)	0.711(3)	0.557(2)	0.599(4)	2.22
SEI-MLHN	0.353(3)	0.722(2)	0.772(1)	0.550(3)	0.798(1)	0.635(1)	0.713(2)	0.573(1)	0.643(2)	1.78

表 6 SI-MLHN 和 SEI-MLHN 在数据集 nus-wide-full-cVLADplus 和 nuswide-bow 上的性能指标(均值±方差)

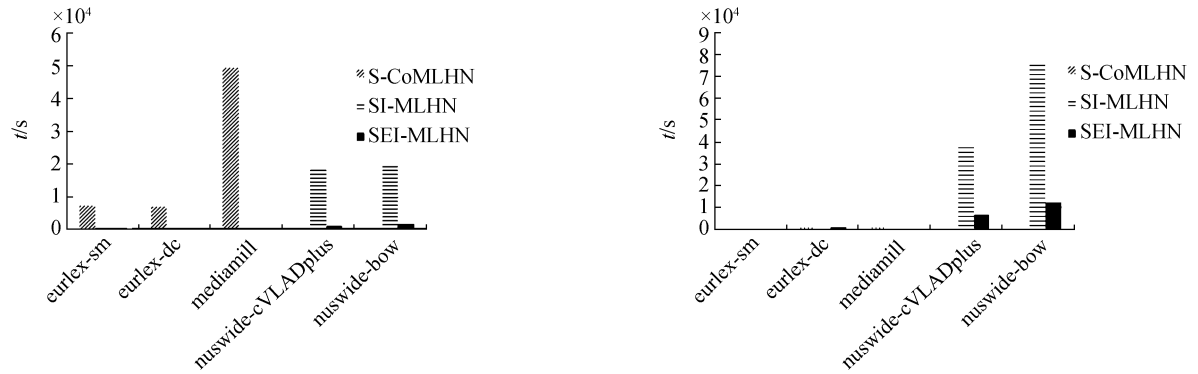
Table 6 Classification performance (mean±std.) of SI-MLHN and SEI-MLHN on nus-wide-full-cVLADplus and nuswide-bow

数据集	算法	Hamming loss	Ranking Loss	One-error	Average Precision	Example-based F_1-measure
nuswide-cVLADplus	SI-MLHN	$(0.0214\pm1.0)\times10^{-4}$	0.1007±0.0318	0.3875±0.0147	0.5183±0.0244	0.3343±0.0184
nuswide-cVLADplus	SEI-MLHN	$(0.021\ 7\pm1.0)\times10^{-4}$	0.110 6±0.002 1	0.390 2±0.001 8	0.515 2±0.001 7	0.348 1±0.002 7
nuswide-bow	SI-MLHN	$(0.023\ 2\pm2.0)\times10^{-4}$	$(0.098\ 8\pm3.0)\times10^{-4}$	0.449 4±0.001 6	0.467 3±0.000	0.310 1±0.003 6
nuswide-bow	SEI-MLHN	0.023 2±0.000	$(0.133\ 7\pm9.0)\times10^{-4}$	0.444 9±0.001 4	$(0.456\ 4\pm5.0)\times10^{-4}$	0.308 7±0.003 1

表 7 各多标签学习算法在不同数据集上的相对性能比较

Table 7 Relative performance comparison of multi-label algorithms on data sets

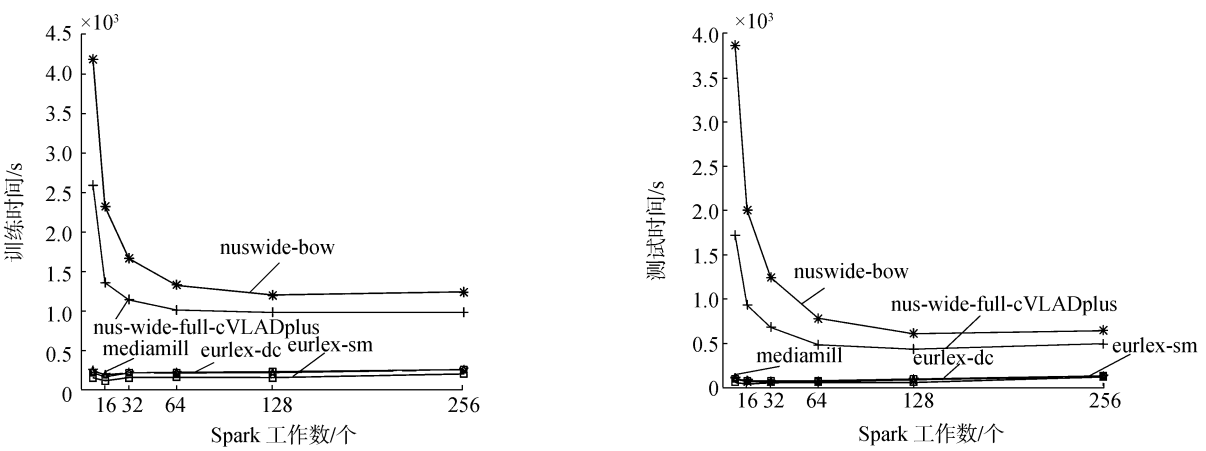
BRSVM(A_1),CLRSVM(A_2),RAkEL(A_3),ECC(A_4),IBLR(A_5),COCOA(A_6),MLKNN(A_7),S-CoMLHN(A_8),SI-MLHN(A_9),SEI-MLHN(A_{10})	
Hamming loss	$A_1>A_2, A_1>A_3, A_4>A_2, A_4>A_3, A_7>A_3, A_8>A_3, A_9>A_3, A_{10}>A_3, A_4>A_6, A_8>A_5,$ $A_9>A_5, A_{10}>_{A5}, A_7>A_6, A_8>A_6, A_9>A_6, A_{10}>A6$
Example based F_1	$A_1>A_3, A_4>A_1, A_6>A_1, A_9>A_1, A_{10}>A_1, A_4>A_2, A_5>A_2, A_6>A_2, A_8>A_2, A_9>A_2, A_{10}>A_2, A_4>A_3, A_5>A_3,$ $A_6>A_3, A_9>A_3, A_{10}>A_3, A_4>A_7, A_{10}>A_4\ A_6>A_5, A_9>A_5, A_{10}>A_5, A_6>A_7, A_9>A_7, A_{10}>A_7, A_9>A_8, A_{10}>A_8$
Ranking loss	$A_2>A_1, A_4>A_1, A_5>A_1, A_6>A_1, A_7>A_1, A_8>A_1, A_9>A_1, A_{10}>A_1, A_2>A_3, A_4>A_3, A_5>A_3, A_6>A_3, A_7>A_3, A_8>A_3,$ $A_9>A_3, A_{10}>A_3, A_6>A_4, A_8>A_4, A_9>A_4, A_{10>}A_4, A_6>A_5, A_8>A_5, A_9>A_5, A_{10}>A_5, A_8>A_{10}$
one-error	$A_1>A_3, A_4>A_1, A_6>A_1, A_7>A_1, A_8>A_1, A_9>A_1, A_{10}>A_1, A_9>A_2, A_{10}>A_2, A_4>A_3, A_5>A_3, A_6>A_3, A_7>A_3, A_8>A_3,$ $A_9>A_3, A_{10}>A_3, A_4>A_5, A_6>A_5, A_9>A_5, A_{10}>A_5, A_{10}>A_8, A_{10>}A_9$
Average precision	$A_1>A_3, A_4>A_1, A_5>A_1, A_6>A_1, A_7>A_1, A_8>A_1, A_9>A_1, A_{10}>A_1, A_8>A_2, A_9>A_2, A_{10}>A_2, A_4>A_3, A_5>A_3, A_6>A_3, A_7>A_3,$ $A_8>A_3, A_9>A_3, A_{10}>A_3, A_8>A_4, A_9>A_4, A_{10}>A_4, A_6>A_5, A_8>A_5, A_9>A_5, A_{10}>A_5, A_6>A_7, A_{10}>A_7, A_{10}>A_8, A_{10}>A_9$
总分	$A_{10}(27)>A_9(20)>A_8(12)>A_6(11)>A_4(6)>A_7(2)>A_5(-11)>A_2(-11)>A_1(-20)>A_3(-36)$



(a)训练时间变化情况　　(b)测试时间变化情况

图 5　算法对不同规模数据集训练与测试时间

Fig.5　Training and testing time of each comparing algorithm on multilabel data sets



(a)训练时间变化情况　　(b)测试时间变化情况

图 6　SEI-MLHN 在不同 SparkTask 下不同数据集的训练与测试时间的变化

Fig.6　Training and testing time of SEI-MLHN under different values of SparkTask

4　结束语

本文基于 MLHN 提出了一种能有效利用标签相关性处理大数据集 Spark 平台下的改进多标签超网络集成算法 SEI-MLHN。该算法首先引入代价敏感,使其适应不平衡数据集并提升算法性能。然后,在超网络演化学习过程中利用抽样信息,以及修改损失函数,降低算法时间复杂度。最后,进行选择性集成,提高算法性能。在 11 个不同规模的数据集上进行实验,结果表明,该算法具有良好的分类性能、较低的时间复杂度以及良好的处理大规模数据集的能力。

在 SEI-MLHN 中,我们只考虑标签之间的相关性,即标签的同现。然而,标签之间的排他关系对于多标签分类也很重要。在未来,我们将研究如何利用超网络标签之间的包含性和排他性。

参考文献:

[1] GAO S, WU W, LEE C H, et al. A MFoM learning approach to robust multiclass multi-label text categorization [C]//Proceedings of the 21st International Conference on Machine Learning. Canada: ACM Press, 2004: 42.

[2] JIANG J Y, TSAI S C, LEE S J. FSKNN: multi-label text categorization based on fuzzy similarity and k nearest neighbors[J]. Expert systems with applications, 2012, 39(3): 2813-2821.

[3] BOUTELL M R, LUO J, SHEN X, et al. Learning multi-label scene classification ☆ [J]. Pattern recognition, 2004, 37(9):1757-1771.

[4] QI G J, HUA X S, RUI Y, et al. Correlative multi-label video annotation [C]//In Proceedings of the 15th ACM International Conference on Multimedia. Germany: ACM Press, 2007: 17-26.

[5] CESA-BIANCHI N, RE M, VALENTINI G. Synergy of

multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference [J]. Machine learning, 2012, 88(1): 209-241.

[6] ZHANG M L, ZHANG K. Multi-label learning by exploiting label dependency [C]//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. USA: ACM Press, 2010: 999-1008.

[7] TSOUMAKAS G, KATAKIS I, VLAHAVAS I. Mining multi-label data [M]. New York: Springer US, 2009: 667-685.

[8] FüRNKRANZ J, HüLLERMEIER E, MENCíA E L, et al. Multilabel classification via calibrated label ranking [J]. Machine Learning, 2008, 73(2): 133-153.

[9] TSOUMAKAS G, KATAKIS I, VLAHAVAS I. Random *k*-Labelsets for Multilabel Classification [J]. IEEE transactions on knowledge & data engineering, 2010, 23 (7): 1079-1089.

[10] LO H Y, LIN S D, WANG H M. Generalized *k*-labelsets ensemble for multi-label and cost-sensitive classification [J]. Knowledge & data engineering IEEE transactions on, 2014, 26(7): 1679-1691.

[11] HE H, GARCIA E A. Learning from imbalanced data [J]. IEEE transactions on knowledge and data engineering, 2009, 21(9): 1263-1284.

[12] XIOUFIS E S, SPILIOPOULOU M, TSOUMAKAS G, et al. Dealing with concept drift and class imbalance in multi-label stream classification [C]//IJCAI 2011 Proceedings of the International Joint Conference on Artificial Intelligence. Barcelona, Spain, 2011: 1583-1588.

[13] CHARTE F, RIVERA A, del JESUS M J, et al. A first approach to deal with imbalance in multi-label datasets [C]//In Proceedings of the International Conference on Hybrid Artificial Intelligence Systems. Springer Berlin Heidelberg, USA, 2013: 150-160.

[14] ZHANG M L, LI Y K, LIU X Y. Towards class-imbalance aware multi-label learning [C]//Proceedings of the 24th International Joint Conference on Artificial Intelligence. Argentina: AAAI Press, 2015: 4041-4047.

[15] LIU H, LI X, ZHANG S. Learning instance correlation functions for multilabel classification [J]. IEEE transactions on cybernetics, 2017, 47(2): 499-510.

[16] ZHANG M L, WU L. Lift: multi-label learning with label [J]. Pattern analysis & machine intelligence IEEE transactions on, 2015, 37(1): 107-20.

[17] ALALI A, KUBAT M. PruDent: A pruned and confident stacking approach for multi-label classification [J]. IEEE transactions on knowledge & data engineering, 2015, 27 (9): 1-1.

[18] WU Q, TAN M, Song H, et al. ML-forest: a multi-label tree ensemble method for multi-label classification [J]. IEEE transactions on knowledge and data engineering, 2016, 28(10): 1-1.

[19] HUANG J, LI G, HUANG Q, et al. Learning label-specific features and class-dependent labels for multi-label classification [J]. IEEE transactions on knowledge and data engineering, 2016, 28(12): 3309-3323.

[20] WU Q, YE Y, ZHANG H, et al. ML-Tree: a tree-structure-based approach to multilabel learning. [J]. IEEE trans neural netw learn syst, 2014, 26(3): 430-443.

[21] CHARTE F. LI-MLC: A Label Inference Methodology for Addressing High Dimensionality in the Label Space for Multilabel Classification [J]. IEEE trans. neural networks and learning systems, 2014, 25(10): 1842-1854.

[22] MONTAñES E, SENGE R, BARRANQUERO J, et al. Dependent binary relevance models for multi-label classification [J]. Pattern recognition, 2014, 47 (3): 1494-1508.

[23] LO H Y, LIN S D, WANG H M. Generalized *k*-labelsets ensemble for multi-label and cost-sensitive classification [J]. Knowledge and data engineering IEEE transactions on, 2014, 26(7): 1679-1691.

[24] SUN K W, LEE C H, WANG J. Multilabel classification via co-evolutionary multilabel hypernetwork [J]. IEEE transactions on knowledge and data engineering, 2016, 28 (9): 1-1.

[25] LO H Y, WANG J C, WANG H M, et al. Cost-sensitive multi-label learning for audio tag annotation and retrieval [J]. IEEE transactions on multimedia, 2011, 13 (3): 518-529.

[26] OZONAT K, YOUNG D. Towards a universal marketplace over the web: statistical multi-label classification of service provider forms with simulated annealing [C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2009: 1295-1304.

[27] ZHANG M L, ZHOU Z H. A review on multi-label learning algorithms [J]. IEEE transactions on knowledge and dada engineering, 2014, 26(8): 1819-1837.

[28] ZHANG M L, ZHANG K. Multi-label learning by exploiting label dependency [C]//Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. USA: ACM Press, 2010: 999-1008.

[29] ZHANG M L, ZHOU Z H. ML-KNN: A lazy learning approach to multi-label learning [J]. Pattern recognition, 2007, 40(7): 2038-2048.

[30] BOUTELL M R, LUO J, SHEN X, et al. Learning multi-label scene classification [J]. Pattern recognition, 2004, 37(9): 1757-1771.

[31] ELISSEEFF A, WESTON J. A kernel method for multi-labelled classification [C]//In NIPS'01 Proceedings of the

14th International Conference on Neural Information Processing Systems: Natural and Synthetic. Vancouver, British Columbia, Canada: MIT Press, 2001:681-687.

[32] ZHANG M L, ZHOU Z H. Multilabel neural networks with applications to functional genomics and text categorization [J]. IEEE transactions on knowledge and data engineering, 2006, 18(10): 1338-1351.

[33] READ J, PFAHRINGER B, HOLMES G, et al. Classifier chains for multi-label classification[J]. Machine learning, 2011, 85(3): 254-269.

[34] YI L, RONG J, LIU Y. Semi-supervised Multi-label Learning by Constrained Non-negative Matrix Factorization. [C]//In AAAI'06 Proceedings of the 21st national conference on Artificial intelligence. Boston: AAAI Press, 2006:421-426.

[35] LIU X Y, LI Q Q, ZHOU Z H. Learning imbalanced multi-class data with optimal dichotomy weights [C]// Proceedings of the 2013 IEEE 13th International Conference on Data Mining. USA: IEEE Press, 2013: 478-487.

[36] TAHIR M A, KITTLER J, MIKOLAJCZYK K, et al. Improving multilabel classification performance by using ensemble of multi-label classifiers[C]//Proceedings of the International Workshop on Multiple Classifier Systems. Egypt: Springer Berlin Heidelberg, 2010: 11-21.

[37] DEAN J, GHEMAWAT S. MapReduce: Simplified data processing on large clusters[C]//In OSDI'04 Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation. Berkeley, USA, 2004: 10-10.

[38] ZAHARIA M, CHOWDHURY M, FRANKLIN M J, et al. Spark: cluster computing with working sets [C]//In HotCloud'10 Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing. Berkeley, USA, 2010: 10-10.

[39] MIKA P. Flink: semantic web technology for the extraction and analysis of social networks[J]. Web semantics science services and agents on the world Wide Web, 2005, 3 (2/3): 211-223.

[40] BU Y, HOWE B, BALAZINSKA M, et al. HaLoop: efficient iterative data processing on large clusters [J]. Proceedings of the Vldb endowment, 2010, 3 (1/2): 285-296.

[41] ZAHARIA M, CHOWDHURY M, DAS T, et al. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing[C]//In Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. San Jose: USENIX Association, 2012: 2.

[42] VESANTO J, ALHONIEMI E. Clustering of the self-organizing map[J]. IEEE transactions on neural networks, 2000, 11(3): 586-600.

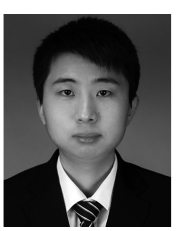
[43] LEE S J, JIANG J Y. Multilabel text categorization based on fuzzy relevance clustering [J]. IEEE transactions on fuzzy systems, 2014, 22(6): 1457-1471.

[44] CHENG W, HüLLERMEIER E. Combining instance-based learning and logistic regression for multilabel classification [J]. Machine learning, 2009, 76(2):211-225.

作者简介:

李航, 女,1995 年生,硕士研究生,主要研究方向为机器学习与数据挖掘。

王进,男,1979 年生,教授,博士,主要研究方向为大数据并行处理与分布式计算、大规模数据挖掘与机器学习。曾主持多项国家和重庆市科研课题,发表学术论文 50 多篇,其中被 SCI 检索 10 篇,授权专利 13 项。



赵蕊,男,1990 年生,硕士研究生,主要研究方向为机器学习与数据挖掘。发表学术论文 2 篇,均被 EI 检索。