计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.190300150

大数据环境下基于关联规则的多标签学习算法

王青松　姜富山　李　菲
辽宁大学信息学院　沈阳 110036

摘　要　传统单标签挖掘技术研究中,每个样本只属于一个标签且标签之间两两互斥。而在多标签学习问题中,一个样本可能对应多个标签,并且各标签之间往往具有关联性。目前,标签间关联性研究逐渐成为多标签学习研究的热门问题。首先为适应大数据环境,对传统关联规则挖掘算法 Apriori 进行并行化改进,提出基于 Hadoop 的并行化算法 Apriori_ING,实现各节点独立完成候选项集的生成、剪枝与支持数统计,充分发挥并行化的优势;通过 Apriori_ING 算法得到的频繁项集和关联规则生成标签集合,提出基于推理机的标签集合生成算法 IETG。然后,将标签集合应用到多标签学习中,提出多标签学习算法 FreLP。FreLP 利用关联规则生成标签集合,将原始标签集分解为多个子集,再使用 LP 算法训练分类器。通过实验将 FreLP 与现有的多标签学习算法进行对比,结果表明在不同评价指标下所提算法可以取得更好的结果。

关键词:多标签学习;LP;关联规则;Apriori;Hadoop
中图法分类号　TP301

Multi-label Learning Algorithm Based on Association Rules in Big Data Environment

WANG Qing-song,JIANG Fu-shan and LI Fei
College of Information,Liaoning University,Shenyang 110036,China

Abstract　In the traditional single-label mining technology research,each sample belongs to only one label and the labels are mutually exclusive. In the multi-label learning problem,one sample may correspond to multiple labels,and each label is often associated with each other. At present,the research on the correlation between tags gradually becomes a hot issue in multi-label learning research. Firstly,in order to adapt to the big data environment,the traditional association rule mining algorithm Apriori is parallelized and improved. The Hadoop-based parallelization algorithm Apriori_ING is proposed to realize the generation of the candidate set,the pruning and the support number statistics,and the parallelization. The advantage is that the frequent itemsets and association rules obtained by the Apriori_ING algorithm generate tag sets,and the inference engine based tag set generation algorithm IETG is proposed. Then,the label set is applied to multi-label learning,and a multi-label learning algorithm FreLP is proposed. FreLP uses association rules to generate a set of labels,decomposes the original set of labels into multiple subsets,and then uses the LP algorithm to train the classifier. FreLP was compared with the existing multi-label learning algorithms. Experiment results show that the proposed algorithm can obtain better results under different evaluation indicators.

Keywords　Multi-label learning,LP,Association rule,Apriori,Hadoop

1　引言

传统二分类和多分类认为真实世界的每一个对象只对应一个类别标记。然而,真实世界中的对象往往具有多义性,可能对应多个类别标记。例如,在图像分类中,一张图片往往可以对应多个标记;一篇报道也可能有多个主题。随着时代发展,在数据量不断增大的同时,数据的复杂程度也在增加,传统的单标签学习已经不能满足技术的发展需要,多标签学习逐渐成为了研究的热点[1]。近年来,多标签学习的研究成果已被大量应用于文本分类[2-4]、音乐情感分类[5-6]、图像视频标注[7-8]、生物信息[9-10]等领域。

目前对多标签学习算法的研究主要集中在两个方面:问题转换和算法适应。问题转换的思路是将多标签学习问题转化为传统的单标签学习问题,比较经典的算法有 BR[11],LP[12]和 CC[13]。BR 算法学习多个分类器而忽略了标签间的关系,算法简单但性能难以令人满意。LP 算法考虑了标签的关联性,但开销较大。CC 算法则用多个分类器构造链式结构,可以有效地利用标签间的关系,但需要额外构造链结构。算法适应是对传统的单标签学习算法进行改进,使其具有处理多标签学习问题的能力,比较常见的算法有基于 Boosting

到稿日期:2019-03-28　返修日期:2019-08-12　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61802160)
This work was supported by the National Natural Science Foundation of China (61802160).
通信作者:王青松(1301833668@qq.com)

的算法 AdaBoost.MH 和 AdaBoost.MR[14]，以及基于决策树的算法。其中，Adaboost.MH 算法的主要思想是最小化汉明损失(Hamming Loss)，Adaboost.MR 的算法思路是最小化排序损失(Ranking Loss)；对于基于决策树的算法，Clare 等[15]改进了经典单标签学习算法 C4.5，使改进算法中的叶节点不再是一个样本而是一个标签集合，同时修改了算法的信息熵计算公式。但这些算法没有充分考虑标签间的关联性，性能一般。

随着物联网和社交网络等新技术和服务的不断发展，如何在大数据环境下挖掘数据之间的关联性早已成为研究的热点[16-18]，而本文的研究重点便是将大数据中蕴含的关联性与多标签学习相结合。

本文首先针对目前关联规则挖掘算法需要适应大数据环境的情况，提出了基于 Hadoop 的 Apriori 算法的并行化算法 Apriori_ING；然后，提出了将关联规则与多标签学习相结合的 FreLP 算法，FreLP 算法利用 Apriori_ING 算法挖掘得到标签之间的关联性，通过合理利用标签间的关联性来减少类别种类，降低算法搜索空间的规模，进而提升算法的准确性。

2 相关工作

2.1 多标签学习的关联性研究

标签间的关联性研究为多标签学习提供更多的信息，能够提升多标签学习算法的性能。因此，如何发掘标签之间的关联性同时有效地加以利用是当前多标签研究领域的核心问题之一。按照对关联性的应用可将多标签学习算法分为一阶策略、二阶策略和高阶策略[19]。

一阶策略选择忽略标签间的关联性，该策略简单且容易实现，但泛化能力较低。比较有代表性的算法是 BR 算法[11]。二阶策略主要考虑了成对标签之间的关系。其性能通常强于一阶策略，但不能全面地解决二阶以上的标签之间的关联。比较有代表性的算法是 Fürnkranz 等[20]提出的 CLR 算法。高阶策略由于考虑了多标签之间的关联性而性能较好，但可能带来计算复杂度过高的问题。比较有代表性的算法是 LP 算法[12]和 Tsoumakas 等[21]提出的 RAkEL。

2.2 关联规则挖掘

关联规则挖掘是指以某种方式分析数据源，从中发现一些潜在的有用信息，因此数据挖掘又称作知识发现。而关联规则挖掘则是数据挖掘中的一个很重要的课题，顾名思义，它是从数据背后发现事物之间可能存在的关联或者联系[22]。其中，Apriori 算法[23-24]是最具影响力的挖掘频繁项集的经典算法之一，其思想是使用逐层迭代的形式，由阶频繁项集生成 $k+1$ 阶候选项集，扫描事务集得到 $k+1$ 阶频繁项集。扫描一遍完整事务集才能得到一个频繁项集。

Apriori 算法的步骤[25]如下：

(1)设 C 为候选项集，L 为频繁项集。遍历原始事务集，将所有出现在事务集中的项作为候选 1-项集。遍历事务集，统计候选项集的支持度。删除所有支持度低于阈值的项，生成频繁 1-项集 L_1。

(2)使用连接步生成待剪枝候选项集，并用剪枝步剪枝得到候选 2-项集 C_2。

(3)再次遍历事务集，得到每个候选集合的支持度。删除所有支持度低于阈值的项，最终得到频繁 2-项集 L_2。

(4)迭代执行上述过程，直到候选集合 C_k 为空为止。

3 基于 Hadoop 的并行化算法 Apriori_ING

本文提出了一种基于 Hadoop 的 Apriori 并行化算法 Apriori_ING。常见的 Apriori 并行化算法主要利用各节点并行化统计候选项集支持数的方式，来适应大数据背景下的频繁项集挖掘。但是由低阶频繁项集到高阶候选项集的生成、剪枝步骤却是在单机上完成的，这使得候选项集的生成、剪枝步骤成为了计算瓶颈。另外，各节点会接收所有的候选项集，所以在统计支持度时，即使某些候选项集中的项并不存在于此节点，该节点还是要遍历一遍事务集为之统计支持数，这一过程将浪费计算资源。Apriori_ING 则针对上述问题进行改进，使各个节点生成其独有的候选项集，提高了候选项集的生成、剪枝和支持数计数的效率。

3.1 Apriori_ING 算法描述

定义 1 设事务集 $D=\{T_1,T_2,\cdots,T_m\}$。其中 T_i 表示第 i 个事务，其由项集中的多个项组成。事务集 D 可以转换成以项 I 为行向量的形式 $D=\{I_1,I_2,\cdots,I_n\}$。其中 I_i 表示事务集中第 i 个项，其由多个事务 T 组成。

定义 2 设 T_x 是项 I_i 的事务集合，T_y 是项 I_j 的事务集合，则计算项集 $\{I_i,I_j\}$ 的支持度的公式为：

$$Support\{I_i,I_j\}=\sum_{k=1}^{m}Tx_k\wedge Ty_k$$

同理，多项集的支持度计算公式为：

$$Support\{I_i,I_j,\cdots,I_n\}=\sum_{k=1}^{m}Tx_k\wedge Ty_k\wedge\cdots\wedge Tz_k$$

定理 1[24] 若 k 维的项集 $X=\{x_1,x_2,\cdots,x_k\}$ 中存在项 $I\in X$，且此项 I 在 $k-1$ 项频繁项集中出现的次数 $|L_{k-1}(I)|<k-1$，则 X 不可能是频繁项集。其中，$|L_{k-1}(I)|$ 表示所有的 $k-1$ 维频繁项集的组成集合 L_{k-1} 中包含项 I 的频繁项集的个数。

定理 2[24] k 维数据项集 X 是频繁项集的必要条件是它的所有 $k-1$ 维子集均是频繁项集。

Apriori_ING 算法利用了定义 1 中的转换方法所带来的分布式候选项集在生成与剪枝方面的优势。以项为单位将事务集 $D=\{I_1,I_2,\cdots,I_n\}$ 分成 m 块不相交的数据子集 d_k，$k=1,\cdots,m$。在候选项集生成过程中，各节点数据子集只需要考虑自身项集中所包含的项，因此只生成较少的候选项集，然后使用定义 2 的方法完成支持度的统计。但是直接使用此种方式会面临一个严重的问题：当出现跨节点的候选项集时，$I_1\in d_i$，$I_2\in d_j$，统计其支持度 $Support\{I_1,I_2\}$ 将变得非常复杂，因为统计时需要计算事务的交集，所以只能令不同节点通信以完成支持度的计数，若出现需要跨多节点统计的候选项集，

则算法总体的效率必然会急剧降低。

针对上述问题，本文先以事务为单位分块，再按照定义1的转换形式，同时增加一个标记位来标记不会再被用到的项。因为事务集的分块操作是根据事务进行分割的，每个节点分得的事务集只看作一个小事务集即可，其本质上与原始事务集只有规模大小的区别。因此按照事务分块保证了每个节点统计候选项集支持度时，只统计自身包括的事务，而无须考虑跨节点统计。但分块后各数据块 $d_k, k=1,\cdots,m$ 不再互不相交，这样算法就以部分冗余项出现在不同节点为代价，来保证在统计支持度时无须进行跨节点统计。

算法 1 Apriori_ING 算法

输入：事务集 D，节点数 m，支持度阈值 s

输出：满足最小支持度的频繁项集

1. 使用 HDFS 机制将事务集分成 n 个大小相似的数据块，并将这些数据块分配到 m 个节点上，执行一次 Mapreduce 程序，实现定义 1 的事务集格式转换。转换过程只需要实现 Map 函数，其伪代码如下：

```
Map:
    Mapper(key,value=Items)
    transFormation(key,Items) //实现定义 1 的事务集格式转换
    Output(Item,Tset)
```

2. 执行 Map 函数，各节点根据定理 1 更新标记位。再对频繁项集进行一次筛选，去除自身事务集中没有的项或者标记位为 0 的项。利用筛选后的频繁项集生成候选项集并利用定理 2 进行剪枝。

3. 统计支持数，生成 pair〈Itemset(项集)，number(支持数)〉。combiner 将各个数据块中 Itemset 相同的 value 值进行局部累加。再经过 shuffle 洗牌和 hash 分桶后由 reduce 函数负责累加全部节点产生的支持度，与 min_sup 做比较后得到频繁项集。

4. 将上一轮产生的频繁项集保存到 HDFS，然后广播到各个节点上。

```
Map:
  Mapper(key,values=Tset)
  proPruning(L_{k-1})//去除自身事务集中没有的项
  aprioriGen(L_{k-1})//连接步生产候选项集 C_k
  pruningStep(C_k)//剪枝步对 C_k 进行剪枝
  supportCount(C_k)//统计 C_k 的支持数
  Output〈Itemset,Itemset_count〉
Reduce:
  Reducer(key,values=itemset_count)
    Foreach count itemset_count
      Sum+=count
    If sum>=s
    Output〈Itemset,sum〉
```

5. 重复执行步骤 2－步骤 4 生成频繁项集，直到生成的 k 项频繁项集的个数小于 k+1 时停止迭代。

Apriori_ING 算法将传统的 Apriori 算法应用于 Hadoop 框架，使其具备在大数据环境下挖掘关联规则的能力；同时，对事务集分块、候选项集的生成与剪枝策略进行了改进。新算法摆脱了常见 Apriori 并行化算法中由单节点生成候选项集，再交由集群并行计算支持数的模式。其将原始事务集分割成包含不同项的数据子集，利用各节点的项不同这一特性可以使每个节点都生成独有的候选项集，可利用频繁项集对候选项集剪枝，减小了生成的候选项集的规模，提高了支持数的统计速度，同时有效提升了各个节点的候选项集的生成、剪枝效率。

3.2 Apriori_ING 算法的复杂度分析

传统 Apriori 算法的时间复杂度为 $O(M^k)$，其中 M 是事务的总数，k 是频繁项集的长度。在 MapReduce 并行框架下，事务集的大小对通信时间有轻微影响，定义通信时间为 t。Apriori_ING 算法对事务集进行分割，各节点只需要考虑自身事务集中包含的项。设集群中节点的数量为 n，则算法的时间复杂度为 $O\left(\left(\frac{\sum_{i=1}^{n}|M_i|}{n}\right)^k\right)+t$。可以看出，当事务集规模较大时，Apriori_ING 的时间复杂度明显比 $O(M^k)$ 小得多，因此在处理海量数据时，Apriori_ING 算法更有效率。

4 基于关联规则的多标签学习算法 FreLP

设有 N 个样本的单标记分类任务中每一个样本都有单一的类别 $l_i, l_i \in L, i=1,\cdots,N$，其中 L 是类别的有限集合。而多标签分类中每一个样本都对应一个类别集合 $Y_i \subseteq L$。现有的多标签分类算法往往是将一个多标签分类问题转变为多个单标签分类问题，这种方法会忽略标签之间的关联性。本文从关联规则入手，在经典多标签学习算法 LP 的基础上提出了一种基于关联规则的多标签分类算法 FreLP。

4.1 基于推理机的标签集合生成算法 IETG

IETG 算法通过对频繁项集和关联规则进行处理，来达到将标签间关联规则与多标签学习算法相结合的目的。处理的思路是：使用这些频繁项集和关联规则主动生成内部具有关联性的多个标签集合。

IETG 的算法流程类似专家系统的推理机模式，如图 1 所示。该算法在频繁 1-项集中选择一个项（通常选用字典顺序得到的第一个项），以这个项为算法的开始并将其作为条件集合，在全体关联规则中寻找满足条件集合的规则，并将规则的结论取出；然后将得到的结论与条件集合合并，形成新的条件集合；使用新的条件集合再次在全体关联规则中寻找满足条件的规则。IETG 算法采用深度遍历的思想，一直循环上述过程，直到没有成立的关联规则为止。每次规则匹配失败后，将之前所有参与循环的项组成一个标签集合。

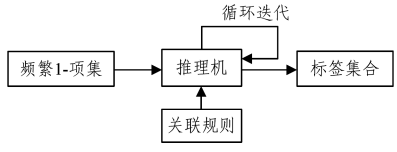

图 1 推理机工作流程图

Fig. 1 Flow chart of inference engine

IETG 算法的基本流程如下：

(1)读取频繁 1-项集，并选取频繁 1-项集中字典顺序的第一个项作为启动项。

(2)将选好的项聚合成一个集合，并将其作为关联规则的条件集合。然后，在所有关联规则中寻找条件相匹配的关联规则（在关联规则生成和匹配过程中，只生成和使用结论为单个项的关联规则）。

(3)匹配到关联规则后，取出关联规则结论中的项，并将提取出的项与现有的条件集合合并，从而形成新的条件集合。然后，将使用过的关联规则从整个关联规则集合中去除。

(4)在全体关联规则中寻找与新产生的条件集合匹配的关联规则。

(5)重复步骤(2)－步骤(4)，直到没有匹配的关联规则为止。

(6)将所有参与循环的项作为一个标签集合输出，然后从频繁 1-项集中选取第二个项作为下一轮迭代的起始项，并重复步骤(2)－步骤(5)，直到频繁 1-项集中的所有项都参与了标签集合的生成。算法结束，得到所有的标签集合。

4.2 FreLP 算法设计

FreLP 算法的详细描述：设在有 N 个样本的单标记分类任务中，每个样本都有单一的标签 I_i，$I_i \in L$，L 是标签的有限集合；而多标签分类中每一个样本都对应一个标签集合。FreLP 算法首先使用 Apriori_ING 挖掘出所有的频繁项集，并由频繁项集利用 IETG 算法生成 m 个不同的标签集合 R_j，$j=1,\cdots,m$。将每个标签集合作为一个新类，每个新类对应的数据集用 D_j 表示，$D_j=\{(x_i, Y_i \cap R_j), i=1,\cdots,N\}$，再使用 LP 算法为每个新类训练分类器 h_j。

算法 2 FreLP 算法

```
输入：数据集 D，频繁 1-项集 L1，关联规则
输出：多标签分类算法模型 hi
For i=1 to |L1| do
/*为频繁 1-项集中的所有项都生成图，然后得到对应的标签集合*/
  R=IETG(L1(i))
  //从频繁 1-项集中的第 i 项得到标签集合。
  m=num(R)
  //m 代表 R 中标签集合的数量
  For i=1 to m do
    Ri←Choose(R)
    //在所有类别集合 R 中挑选一个类别集合。
  Train hi using Ri and D
    /*使用 LP 算法在数据集 D 和数据类别集 Ri 上训练学习器 hi*/
    R←R\Ri
  End For
End For
```

FreLP 算法中使用的标签集合是由频繁项集和关联规则生成的，标签集合生成算法保证了标签集合内部的标签之间具有关联性，FreLP 算法正是利用标签间的关联性来提高算法的性能，减小了算法搜索空间的规模。通过标签集合生成算法可以生成原数据集中没有的标签集合，增强了对样本中未出现的标签集合的泛化能力。另外，具有较强关联性的标签更容易出现在同一样本中，如一张图中同时有蓝天、椰树、遮阳伞等标签是常见的，而鲸鱼和灌木丛很少出现在同一张图片中。因此，内部标签间具有关联性的标签集合对应的样本数量往往较多，可以缓解多标签学习领域中经常出现的样本稀疏问题。

4.3 FreLP 算法的复杂度分析

设单标签学习算法的时间复杂度为 $O(g(C,N,A))$，其中 C 是类的数量，N 是样本数量，A 是预测属性数量。那么 FreLP 算法的时间复杂度为 $O(\hat{m}g(\min(N,2^{\hat{k}}),N,A))$，其中 $\hat{m}$ 是由频繁项集生成的类别集合个数的平均值，$\hat{k}$ 是类别集合的平均长度。FreLP 算法和 RAkEL 算法在时间复杂度分析方面类似，算法 RAkEL 中有标签集合规模 k 和标签集合数量 m 这两个超参数，k 和 m 决定了 RAkEL 的算法复杂度，只要挑选合适的超参数，时间复杂度就会在可接受的范围。FreLP 算法中类别集合的规模和数量是由关联规则产生的，但是在得到关联规则的过程中还需要设定两个超参数：支持度(Support)和置信度(Confidence)。FreLP 算法通过设定支持度和置信度间接影响算法的复杂度。因此，在选用合适的超参数的情况下，相比 LP 算法，两者的时间复杂度属于同一数量级且处于可接受的范围。

5 实验结果和分析

实验环境采用 13 台计算机在 Linux 环境下搭建 Hadoop 集群，采用 Ubuntu 操作系统。JDK 版本为 1.7。其中一台计算机作为 NameNode 不参与运算，其他作为 DataNode，每个节点的 CPU 为 Intel 酷睿 i5，4 GB 内存。采用的数据集分别是歌曲分类领域的 Emotions、基因检测领域的 Yeast、图像语义检索领域的 Scene、文本分类领域的 Medical 和蛋白质分类领域的 Genbase。实验数据皆可在 Mulan[26] 开源项目中下载。表 1 列出了实验数据集的详细信息。

表 1 数据集的详细信息

Table 1 Details in datesets

数据集	样本个数	标签个数	离散值	连续值	平均标签数
Emotions	593	6	—	72	1869
Yeast	1500	14	—	103	4237
Scene	2407	6	—	294	1074
Medical	978	45	1449	—	1245
Genbase	662	27	1186	—	1252

在 Hadoop 集群下运行 Apriori_ING 算法得到频繁项集，将频繁项集和由频繁项集得到的关联规则作为 FreLP 算法的输入，并运行 FrelP 算法。为了验证 FreLP 算法的性能，本文将 FreLP 算法与常见的多标签学习算法(RAkEL 算法、MLKNN 算法、CC 算法、BR 算法)做作比较。同时使用 Accuracy，Recall，F-measure，Ranking Loss 作为算法性能的评价标准。实验结果如表 2－表 5 所列，结果均以均值±标准差的形式给出。

表 2 各算法准确度的比较

Table 2 Accuracy comparison of algorithms

	FreLP	RAkEL	MLKNN	CC	BR
Emotions	**0.5799±0.0377**	0.5710±0.0231	0.5690±0.0401	0.5677±0.0236	0.5521±0.0376
Yeast	**0.6342±0.0157**	0.6245±0.0272	0.6112±0.0371	0.6018±0.0080	0.5974±0.0144
Scene	**0.7220±0.0546**	0.7011±0.0324	0.6954±0.0928	0.6912±0.0462	0.6588±0.0114
Medical	0.7745±0.0036	0.7731±0.0123	**0.7769±0.0520**	0.7571±0.0147	0.7456±0.0324
Genbase	**0.9810±0.0671**	0.9761±0.0652	0.9658±0.0357	0.9747±0.0167	0.9710±0.0673

表 3 各算法召回率的比较

Table 3 Recall comparison of algorithms

	FreLP	RAkEL	MLKNN	CC	BR
Emotions	**0.6581±0.0237**	0.6510±0.0134	0.5730±0.0311	0.5590±0.0283	0.5521±0.0376
Yeast	**0.7112±0.0231**	0.7001±0.0222	0.6955±0.0131	0.6821±0.0190	0.6974±0.0010
Scene	**0.6970±0.0721**	0.6611±0.05421	0.6766±0.0638	0.6891±0.0697	0.6532±0.0122
Medical	0.8411±0.0147	0.8423±0.0652	**0.8658±0.0930**	0.8567±0.0617	0.7956±0.0111
Genbase	0.9612±0.0782	**0.9711±0.0698**	0.9600±0.0431	0.9610±0.0274	0.9531±0.0555

表 4 各算法 F-measure 的比较

Table 4 F-measure comparison of algorithms

	FreLP	RAkEL	MLKNN	CC	BR
Emotions	**0.6897±0.0666**	0.6147±0.0689	0.6711±0.0011	0.5990±0.0349	0.5521±0.0140
Yeast	**0.5216±0.0642**	0.5125±0.0867	0.4877±0.0146	0.5180±0.0633	0.5421±0.0164
Scene	**0.7240±0.0256**	0.7163±0.0964	0.7022±0.0445	0.6853±0.0145	0.6846±0.0485
Medical	**0.7766±0.0633**	0.7651±0.0345	0.7611±0.0010	0.7150±0.0755	0.6956±0.0345
Genbase	0.9764±0.0131	**0.9862±0.0335**	0.9811±0.0317	0.9766±0.0369	0.9711±0.0313

表 5 各算法 Ranking Loss 的比较

Table 5 Ranking loss comparisons of algorithms

	FreLP	RAkEL	MLKNN	CC	BR
Emotions	**0.1755±0.0177**	0.2100±0.0221	0.1990±0.0101	0.2235±0.0163	0.1991±0.0110
Yeast	**0.1549±0.0122**	0.1666±0.0100	0.1648±0.0173	0.1784±0.0180	0.1597±0.0122
Scene	**0.0878±0.0021**	0.1201±0.0110	0.0988±0.0082	0.1455±0.0121	0.1088±0.0191
Medical	**0.0765±0.0036**	0.0825±0.0023	0.1051±0.0610	0.0988±0.0147	0.0879±0.0210
Genbase	**0.2100±0.0101**	0.2218±0.0164	0.3140±0.0222	0.2685±0.0155	0.2498±0.0201

实验中，将 FreLP 算法的支持度和置信度分别设置为 0.1 和 0.7。RAkEL 算法的超参数 $k=3$，$m=2$，阈值 $t=0.5$。MLKNN 中 $k=9$，$Smoothing=1$。FreLP 和 RAkEL 的基础分类器选用 LP，BR 及 CC 算法中的基础分类器选用 SVM。

从实验结果来看，在不同评价标准下，FreLP 算法在 Emotions，Yeast，Scene，Medical，Genbase 数据集上的表现基本优于其他算法，整体效果超出 5%左右。从表 2－表 5 中可以看出，在 Emotions 和 Scene 数据集上，FreLP 算法的提升最明显，因为其性能与数据集中的标签关联性的强弱相关，标签之间的关联性越强，由 Apriori_ING 挖掘生成的关联规则和由关联规则生成的标签集合对算法性能的提升就越大。

下面通过改变关联规则挖掘时的支持度(Support)和置信度(Confidence)来分析标签之间的关联性对算法性能的影响。实验采用 Emotions 数据集，算法性能指标采用准确度(Accuracy)。

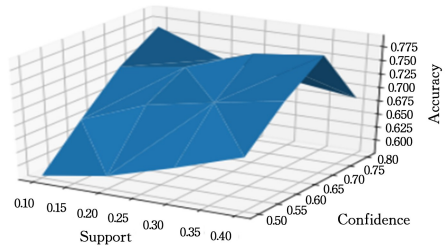

图 2 不同支持度和置信度下的算法性能对比

Fig. 2 Comparison of algorithm performance under different support and confidence

从实验数据中可以看出，关联性对算法的性能有直接影响。在一定范围内，随着支持度与置信度的提升，标签之间的关联性越强，算法性能的提升就越大。当支持度和置信度到达一定程度时，如图 1 中支持度为 0.3、置信度为 0.7 时，算法效果达到最好。

结束语 本文首先提出了一种关联规则挖掘算法 Apriori 的改进算法 Apriori_ING，该算法将 Apriori 与 Hadoop 框架相结合，实现了各节点独立完成候选项集的生成、剪枝和支持数统计，突破了 Apriori 并行化改进的效率瓶颈。然后，将关联规则引入多标签学习算法中，提出了基于推理机的标签集合生成算法 IETG，使用关联规则自动生成多个标签集合。最后，提出多标签学习算法 FreLP，为每个标签集合训练 LP 算法，并使用投票算法汇总所有分类器的结果。在多个数据集上的实验表明，在控制好支持度(Support)和置信度(Confidence)的前提下，引入关联规则到多标签学习任务中可以提高算法的准确性，且算法复杂度可接受。当前的工作只是用关联规则提高多标签学习算法的性能，下一步工作会将关联规则挖掘算法和多标签学习算法融合为一种算法，把支持度和置信度也作为可学习的参数，并找到合适的损失函数。

参 考 文 献

[1] TSOUMAKAS G, KATAKIS I, VLAHAVAS I. Mining multi-labeldata[M]// Data mining and knowledge discovery handbook. US: Springer, 2010: 667-685.

[2] LI L, WANG M, ZHANG L, et al. Learning semantic similarity-for multi-label text categorization[C]// Chinese LexicalSemantics Lecture Notes in Computer Science. 2014: 260-269.

[3] RUBIN T N, CHAMBERS A, SMYTH P, et al. Statistical topic models for multi-label document classification[J]. Machine Learning, 2012, 88(1): 157-208.

[4] 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(1): 521-530.

[5] LIU S M, CHEN J H. A multi-label classification based approach for sentiment classification[J]. Expert Systems with Applications, 2015, 42(3): 1083-1093.

[6] HUANG S, PENG W, LI J, et al. Sentiment and topic analysis on social media: a multi-task multi-label classification approach

[C]// Proceedings of the 5th Annual ACM Web Science Conference. 2013:172-181.

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

[8] WU B, LYU S, HU B G, et al. Multi-label learning with missing labels for image annotation and facial action unit recognition[J]. Pattern Recognition, 2015, 48(7):2279-2289.

[9] ZHANG M L, ZHOU Z H. Multi-label neural networks with applications to functional genomics and text categorization [J]. IEEE Transactions on Knowledge and Data Engineering, 2007, 18(10):1338-1351.

[10] ZHOU Y, XUE H, GENG X. Emotion distribution recognition from facial expressions[C]// Proc. of the ACM Int'l Conf. on Multimedia. 2015:1247-1250.

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

[12] READ J, PFAHRINGER B, HOLMES G. Multi-label classification using ensembles of pruned sets[C]// 8th IEEE International Conference on Data Mining (ICDM'08). 2008:995-1000.

[13] READ J, PFAHRINGER B, HOLMES G, et al. Classifier chains for multi-label classification[C]// 20th European Conference on Machine Learning(ECML'09). Berlin:Springer, 2009:254-269.

[14] SCHAPIRE R E, SINGER Y. BoosTexter:a boosting-based system for text categorization[J]. Machine Learning, 2000, 39(2/3):135-168.

[15] DOQUIRE, GAUTHIER, VERLEYSEN, et al. Mutual information-based feature selection for multilabel classification [J]. Neurocomputing, 2013, 122:148-155.

[16] LI S N, LI N, LI Z H. Multi-label Data Mining Technology: A Review [J]. Computer Science, 2013, 40(4):14-21.

[17] LIU J Y, JIA X Y. A multi-label classification algorithm using association rules mining [J]. Journal of Software, 2017, 28(11):2865-2878.

[18] XIAO W, HU J, ZHOU X F. A Survey of Algorithms for Mining Parallel Association Rules Based on MapReduce-based Computing Model [J]. Computer Applied Research, 2018, 35(1):13-23.

[19] ZHANG M L, ZHOU Z H. A Review on Multi-Label Learning Algorithms [J]. IEEE Trans. on Knowledge and Data Engineering, 2014, 26(8):1819-1837.

[20] FURNKRANZ J, HULLERMEIER E, MENCIA E L, et al. Multi-labelclas-sification via calibrated label ranking [J]. Machine Learning, 2008, 73(2):133-152.

[21] TSOUMAKAS G, VLAHAVAS I. Random k-labelsets:an ensemble method for multilabel classification[C]// Proceedings of the 18th European Conference on Machine Learning. 2007:406-417.

[22] CHENG X Q, JIN X L, WANG Y Z, et al. Survey on big data system and analytic technology[J]. Journal of Software, 2014, 25(9):1889-1908.

[23] AGRAWAL R, SRIKANT R. Fast algorithm for mining association rules[C]// Processdings of 20th Int. Conf. Very Large Data Bases(VLDB). Morgan Kaufman Press. 1994:487-499.

[24] XING C Z, AN W G, WANG X. Improvement of algorithm for mining frequent itemsets in vertical data format [J]. Computer Engineering and Science, 2017, 39(7):1365-1370.

[25] LIU S H, LIU S J, CHEN S X, et al. IOMRA:a high efficiency frequent itemset mining algorithm based on the MapReduce computation model[C]// Proc of IEEE International Conference on Computational Science and Engineering. 2014:1290-1295.

[26] TSOUMAKAS G, VILCEK J, XIOUFITS E S. Mulan: A Java library for multi-label learning [OL]. http://mulan.sourceforge.net/datasets.html.

WANG Qing-song, born in 1974, associate professor. His main research interests include big data and data mining.