软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(11):2865−2878 [doi: 10.13328/j.cnki.jos.005341]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种利用关联规则挖掘的多标记分类算法*

刘军煜, 贾修一

(南京理工大学 计算机科学与工程学院,江苏 南京 210094)

通讯作者: 贾修一, E-mail: jiaxy@njust.edu.cn

摘 要: 多标记学习广泛存在于现实生活中,是当今机器学习领域的研究热点.在多标记学习框架中,每个对象由一个示例构成,但可能同时属于多个类别标记,并且各个标记之间相互关联,所以挖掘多标记之间的关联性对于多标记学习框架具有重要的意义.首先对经典的关联规则算法进行改进,提出了基于矩阵分治的频繁项集挖掘算法,并证明了该算法挖掘频繁项集的正确性;进而将该算法应用于多标记学习框架中,分别提出了基于全局关联规则挖掘和局部关联规则挖掘的多标记分类算法;最后对所提出的算法与现有多标记算法进行实验对比,结果表明,算法在 5 种不同的评价准则下能够取得更好的效果.

关键词: 多标记学习;关联规则;矩阵分治;频繁项集
中图法分类号: TP181

中文引用格式: 刘军煜,贾修一.一种利用关联规则挖掘的多标记分类算法.软件学报,2017,28(11):2865−2878. http://www.jos.org.cn/1000-9825/5341.htm

英文引用格式: Liu JY, Jia XY. Multi-Label classification algorithm based on association rule mining. Ruan Jian Xue Bao/ Journal of Software, 2017,28(11):2865−2878 (in Chinese). http://www.jos.org.cn/1000-9825/5341.htm

Multi-Label Classification Algorithm Based on Association Rule Mining

LIU Jun-Yu, JIA Xiu-Yi

(School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract: In the real world, multi-label learning has become a hotspot in machine learning research area. In the multi-label learning problem, each instance is usually described by multiple class labels, which could be correlated with each other. It is well known that exploiting label correlations is important for multi-label learning. In this paper, an improved association rule mining algorithm based is designed on the matrix divide-and-conquer strategy. In addition, a proof is given to show the proposed algorithm in finding correct frequent items, and an application of the algorithm to the multi-label learning framework is also provided. Moreover, a global association rule mining and a local association rule mining based multi-label classification methods are proposed. Experimental results on several datasets show that the proposed methods can obtain a better classification performance on 5 different evaluation criteria.

Key words: multi-label learning; association rule; matrix divide-and-conquer; frequent item

在传统的监督学习框架中,每个示例仅仅对应于唯一一个类别标记,这类问题可称为单标记学习问题.然而在很多现实问题中,一个示例可能并不仅仅由单个标记描述,而是同时拥有多个类别标记.例如,一篇文档可能属于多个预先定义的主题,一张图片可能同时包含多个语义,一个基因可能同时拥有多种功能等.这种单个示例

* 基金项目: 国家自然科学基金(61773208, 61403200, 71671086); 浙江省海洋大数据挖掘与应用重点实验室资助项目(OBDMA201602)

Foundation item: National Natural Science Foundation of China (61773208, 61403200, 71671086); Foundation of Key Laboratory of Oceanographic Big Data Mining and Application of Zhejiang Province (OBDMA201602)

本文由复杂环境下的机器学习研究专刊特约编辑张道强教授推荐.

收稿时间: 2017-04-13; 修改时间: 2017-06-16; 采用时间: 2017-08-23

拥有多个标记的学习问题称为多标记学习[1−3]问题.近年来,多标记学习得到了许多学者的关注,并被广泛研究应用于文本分类[1−3]、图像标注[4,5]、音频类别标注[6]、视频自动注释[7]等多个领域.

在多标记学习问题中,多个标记同时隶属于一个示例,并且这些标记之间存在一定的关联性[8].举例来说,如果一篇新闻文章已经被标记为“姚明”,那么这则新闻也很有可能被同时标记为“篮球”;再比如一幅图像已经被标记为“鲨鱼”,那这幅图像也很有可能被同时标记为“海洋”.因此在多标记学习问题中,如果忽略标记之间的关系,将多标记学习问题转化为多个单标记学习问题,这样虽然能够处理多标记数据,但是会损失较多的标记关系信息,通常分类效果不会太好.此外,如果直接把标记之间的所有组合作为分类结果输出,随着标记数目的增多,输出组合呈指数级别变化,既增加了模型的复杂性,又容易导致样本的稀疏性.

针对上述问题,本文采用关联规则算法挖掘标记之间的关联性,这样既不改变样本分布,又能避免标记增多时的“维数灾难”问题.现有研究虽然将关联规则算法应用到多标记学习中[9−11],但得到的前项是特征,后项是标记的关联规则,然后,通过一定的方式筛选出部分规则并输出到分类器中.这类方法本质上仍是把关联规则算法作为一种单标记学习的分类算法,考虑的是特征与标记之间的关系,而没有从标记之间的关联性入手.此外,由于传统关联规则算法只能挖掘离散型数据之间的频繁项集,当特征值为连续型数据时则无能为力.本文将关联规则算法用于挖掘标记之间的频繁项集,并不涉及示例的特征,由此可以处理不同数据类型的多标记学习问题.

本文首先针对现有经典关联规则算法需要对数据库进行多次扫描确定候选频繁集,从而导致计算频繁项集效率较低等问题,对关联规则算法进行改进,提出了基于矩阵分治的频繁项集挖掘算法(matrix divide-and-conquer based frequent items mining,简称 MDC-FIM),并对算法的正确性予以证明.其次,基于 MDC-FIM 算法,我们提出了一种基于全局关联规则挖掘的多标记分类算法(global association rule based multi-label classification,简称 GAR-MLC).该算法应用 MDC-FIM 算法挖掘标记之间的关联规则,并利用该关联规则更新多标记数据的标记分布,在此基础上,应用现有的多标记分类算法.考虑到实际情况下标记之间的相关性只存在于部分子数据集,比如在美食杂志中,“苹果”和“水果”存在相关性;再比如在科技杂志中,“苹果”和“电脑”存在相关性,GAR-MLC 只从标记空间进行考虑,而不考虑样本空间,对全局数据进行修正可能会引起原先不属于一个类别的数据被强行修正造成分类准确率降低的问题,进而提出了一种基于局部关联规则挖掘的多标记分类算法(local association rule based multi-label classification,简称 LAR-MLC).该算法先使用聚类算法对示例特征聚类,再利用 MDC-FIM 算法在每个聚类类别下挖掘关联规则并更新标记分布,即对多标记数据作局部的修改.该算法既考虑标记之间的相关性,又考虑样本之间的相关性,因此能够更加合理地修正数据,从而达到更好的分类效果.最后,我们在 6 个多标记基准数据集上,基于 5 种不同的多标记评价准则对本文所提算法进行了实验验证,实验结果表明:我们所提出的 MDC-FIM 算法能够快速挖掘标记之间的频繁项集,改进的 GAR-MLC 和 LAR-MLC 算法与现有的多标记算法相比具有更好的分类性能.

本文第 1 节介绍多标记学习和关联规则挖掘的相关概念和研究现状.第 2 节给出基于矩阵分治的频繁项集挖掘算法(MDC-FIM),并对算法的正确性予以证明.第 3 节给出基于全局关联规则挖掘的多标记分类算法(GAR-MLC)和基于局部关联规则挖掘的多标记分类算法(LAR-MLC).第 4 节给出实验设置与结果分析.第 5 节对本文进行总结和展望.

1 相关工作

1.1 多标记学习

目前,许多学者致力于研究多标记学习问题,基于对标记相关性的不同考虑方式,目前的多标记学习算法主要分为以下 3 种.

(1) 一阶策略

该策略通过逐一考察单个标记而忽略标记之间的相关性构造多标记学习算法.这类策略效率较高并且实现简单,但由于完全忽略标记之间可能存在的相关性,泛化性能较差.

文献[12]提出的 BR 算法将多标记学习问题转化为一系列独立的两类问题,其中,对于第 i 个两类问题,如果

样本属于第 i 个标记,则将其分为正类样本;否则,将其分为负类样本.该算法简单易行,但忽略了标记之间的相关信息.

文献[13]将经典的 KNN 方法应用于多标记学习框架,提出了多标记懒惰学习方法(ML-KNN).该算法对每个测试样本根据它与训练样本的相似度来预测所对应的类别标记集,减少大量的训练开销,但是并没有考虑标记之间的相关性.在此基础上,文献[14]提出一种新型多标记学习算法.该算法考虑了标记之间的相关性,将标记信息构造一个标记计数向量,并提交给已训练的线性分类器进行预测.

(2) 二阶策略

该策略通过考察两两标记之间的相关性(例如相关标记与无关标记之间的排序关系等)构造多标记学习算法.这类策略由于在一定程度上考察标记之间的相关性,其系统泛化性能较优;然而当真实世界问题中标记之间具有超过二阶的相关性时,该类方法的效果会受到很大影响.

文献[15]将传统的支持向量机(SVM)拓展到多标记学习中,对每个类别标记构造一个 SVM 分类器,最小化基于排序损失的经验损失函数,提出了 Rank-SVM 算法.该算法的优点是考虑了每个样本中两个类别标记之间的关系,且该优化问题可转化成一个二次规划问题,但由于需要计算大量的变量,训练时间比较久.

文献[16]基于神经网络,通过改进反向传播算法来适应多标记问题,提出了 BP-MLL(back-propagation multi-label learning)算法.该算法主要引入新的误差函数计算方法来捕捉多标记的特征,并且使用梯度下降算法来更新参数.

(3) 高阶策略

该策略通过考察高阶的标记相关性(例如处理任一标记对其他所有标记的影响等)构造多标记学习算法.该类方法虽然可以较好地反映真实世界问题的标记相关性,但其模型复杂度往往过高,以至于难以处理大规模学习问题.

文献[17]提出的 Random k-labelsets(RAkEL)算法能够考察高阶的标记相关性.算法首先从初始的类别标记集中随机地选取包含一个类别标记的子集,然后训练多个多类分类器,最后,利用集成学习的思想采用投票或阈值法来预测样本所属的类别标记集合.

文献[18]将贝叶斯网络应用于多标记学习算法中,提出了基于贝叶斯网络的多标记学习算法 LEAD.算法采用一种近似的方式首先消除特征向量对于标记空间的影响,然后再构造相应的贝叶斯网络来对标记之间的相关性进行建模.

文献[19]认为标记之间的相关性更可能存在于局部数据中,提出了 ML-LOC 算法.该算法首先基于标记向量使用聚类算法定义局部数据,再融合局部标记相关性和全局可分性到一个统一的分类框架中,并给出了相应的优化求解算法.

1.2 关联规则

若两个或多个变量的取值之间存在某种规律性,则称为关联[20].关联规则是寻找在同一个事件中出现的不同项的相关性,比如在一次购买活动中,购买的不同商品之间的相关性.在机器学习研究领域,学者已经提出多种关联规则的挖掘算法,如 Apriori,FP-growth,Eclat[21]算法等.

文献[9]将关联规则应用于分类算法中,提出了经典的基于分类关联规则(CARs)的分类算法.该算法首先挖掘所有规则后项是标记的分类关联规则(CARs),然后在已发现的 CARs 中选择优先级高的规则来覆盖训练集.随后几年时间中,研究人员研究出很多新的关联分类算法,其中最有代表性的是文献[10]提出的基于多个分类关联规则的分类算法 CMAR 和文献[11]提出的预测型关联规则的分类算法 CPAR.上述分类算法具有可解释性好、分类准确率较高的特点,但是都只能对单标记数据进行处理.目前,国内外对关联规则应用于多标记学习中的研究还比较少.文献[22,23]利用最小标记准则将多标记转化为单标记,分别使用 FP-growth 算法和使用改进的 Apriori 算法得到规则前项是示例特征、后项是标记的关联规则,并基于关联规则进行分类预测.文献[24]利用 FP-tree 来分解组合生成的多标记规则,通过集成学习的方法组合多个关联规则分类器并进行分类预测.为了解决大规模数据集上训练时间太长的问题,文献[25]使用 FP-growth 算法得到关联规则,并删除规则后项中的标记,

从而减少标记数量,提高训练效率,但这在一定程度上会造成消息的丢失,降低分类性能.

2 基于矩阵分治的频繁项集挖掘算法

2.1 关联规则定义

假设给定数据集 DB,$Item=\{i_1,i_2,\ldots,i_m\}$是其中所有项的集合.$Tr=\{tr_1,tr_2,\ldots,tr_k\}$表示事务集,其中,每个事务 Tr_i 是一个非空项.若 i_s,i_t 是 $Item$ 中的任意项集且 i_s 中含有的项数目为 n,则称为 n-项集.数据集 DB 中的关联规则表示为蕴涵式 $i_s\Rightarrow i_t$,且 $i_s\subset Item,i_t\subset Item,i_s\cap i_t=\varnothing$.它们是由支持度(support)和置信度(confidence)约束的,支持度表示规则的频度,置信度表示规则的强度.

规则 $i_s\Rightarrow i_t$ 在 DB 中的支持度是指 Tr 中同时包含 i_s,i_t 的交易数与 Tr 的个数之比,记作:

$$Support(i_s\Rightarrow i_t)=|\{Tr:i_s\cup i_t\subseteq Tr,Tr\in DB\}|/|DB| \tag{1}$$

规则 $i_s\Rightarrow i_t$ 在 DB 中的置信度是指 Tr 中同时包含 i_s,i_t 的交易数与 Tr 中包含 i_s 的个数之比,记作:

$$Confidence(i_s\Rightarrow i_t)=|\{Tr:i_s\cup i_t\subseteq Tr,Tr\in DB\}|/|\{Tr:i_s\subseteq Tr,Tr\in DB\}| \tag{2}$$

关联规则挖掘就是在 DB 中找出支持度和置信度分别大于用户给定的最小支持度($Minsup$)和最小置信度($Minconf$)的关联规则.当规则的置信度和支持度分别大于 $Minsup$,$Minconf$ 时,认为该关联规则是有效的,称为强关联规则.

2.2 基于矩阵分治的频繁项集挖掘算法

对于数据集 DB,定义事务集 $Tr=\{tr_1,tr_2,\ldots,tr_k\}$,项全集 $Item=\{it_1,it_2,\ldots,it_m\}$,生成行向量 $V=\{v_1,v_2,\ldots,v_m\}$用于存储不同的项集名称,生成布尔矩阵 M 用于存储数据库的信息.其中,矩阵中的每项定义为

$$M_{ij}=\begin{cases}0, & it_j\notin tr_i\\ 1, & it_j\in tr_i\end{cases},i\in[1,k],j\in[1,m].$$

基于矩阵分治的频繁项集挖掘算法(MDC-FIM)的主要思想是:对布尔矩阵 M 进行初等变换,在满足一定条件的情况下将其分治为若干个子矩阵,并重复上述过程,直至矩阵不能被分治.该算法最大的优点是只需要一遍扫描整个数据库,运行速度较快.

MDC-FIM 算法的具体步骤为:(1) 将当前布尔矩阵 M 按列分块,并统计每列中 0 的个数;(2) 交换 M 中 0 元素最多的列与最后一列,并对 M 进行初等变换,使最后一列先出现 0 再出现 1;(3) 将 M 分治成若干个子矩阵,并记下项集名称 V;(4) 重复第(1)步,直至求出在满足最小支持度 $Minsup$ 下的频繁项集 V 以及其出现的次数(支持数).MDC-FIM 算法的详细流程见算法 1.

算法 1. 基于矩阵分治的频繁项集挖掘算法(MDC-FIM 算法).

输入:数据库 DB,事务集 Tr,项全集 $Item$,最小支持度 $Minsup$,项集名称 V.

输出:满足最小支持度的频繁项集 $FreItem$.

```
1.  Generate M using DB, Tr, Item
2.  j=Find(M)                          //M 中第 j 列 0 元素个数最多
3.  Switch(M[:,j],M[:,end]);           //交换 M 中第 j 列和最后一列
4.  Switch(V[j],V[end]);               //交换 M 中两个项集
5.  Sort(M,end);                       //对 M 进行初等列变换,使第 end 列先出现 0 再出现 1
6.  countend=sum(M[:,end]);            //计算第 end 列值为 1 的个数
7.  threshold=Minsup×row(M);           //设置阈值
8.  if (end>1)
9.     FreItem.add(M[:,1:end−1]);      //增加一个第 1 列到 end−1 列的新矩阵
10.    FreItem.add(V[1:end−1]);        //增加一个第 1 个到 end−1 的新向量
11.    Goto 2);
```

12. endif
13. if (*countend*≥*threshold*)
14. *begin*=*row*(*M*)−*threshold*+1;
15. *FreItem.add*(*M*[*begin*:*row*(*M*),:]); //增加一个 *begin* 到 *row*(*M*)行的新矩阵
16. *FreItem.add*(*V*[1:*end*]); //增加整个 *V* 向量
17. Goto 2);
18. endif
19. return *FreItem*

2.3 MDC-FIM算法推导频繁项集的正确性证明

对于数据集 *DB*,事务集 $Tr=\{tr_1,tr_2,\ldots,tr_k\}$,项全集 $Item=\{i_1,i_2,\ldots,i_m\}$,布尔矩阵为 *M*(*M* 不全为 0),即共有事务个数为 *k*,项个数为 *m*.不妨设其中一列中 0 元素的个数最多,且为 *j* 个,则 *M* 经过初等变换一定可以转化为如图 1 所示的形式,即最后一列顺序由 *j* 个 0 和 *k*−*j* 个 1 构成.

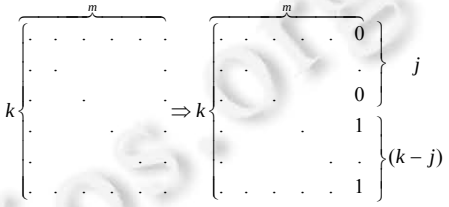

Fig.1 Elementary transformation of matrix

图 1 矩阵进行初等变换

下面证明:原矩阵 *M* 满足最小支持度 *Minsup* 的频繁 *n*-项集若存在,则必在其子矩阵 M_1 或者 M_2 中,如图 2 所示).

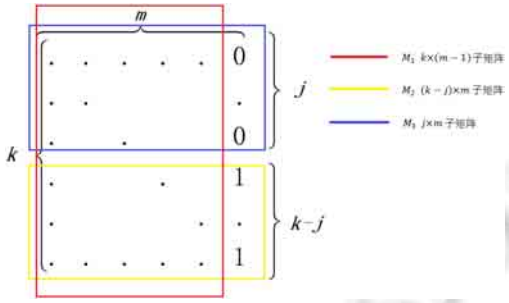

Fig.2 An example of matrix divide-and-conquer

图 2 矩阵分治示例

首先假设频繁 *n*-项集有 *N* 条事务.

假设满足支持度 *Minsup* 的频繁 *n*-项集存在且不在 M_1 和 M_2 中,则有如下两种情况.

1) 满足支持度 *Minsup* 的频繁 *n*-项集在图 2 所示的子矩阵 M_3 中.

这种情况下,M_3 每行的最后一位均为 0,所以其 *n*-项集由前 *m*−1 列产生,而 M_3 的前 *m*−1 列包含于 M_1 中,所以此时 *n*-项集包含于 M_1 中,假设不成立.

2) 满足条件的 *n*-项集有 *a*(0<*a*<*N*)条在 M_2 中,其余 *N*−*a* 条在 M_3 中.

由情形 1)已知,M_3 的解包含于 M_1 的前 *j* 行之中,所以这 *N*−*a* 条事务存在于 M_1 的前 *j* 行之中.由满足支持度条件的频繁 *n*-项集的定义可知,这 *N* 条事务有且只有相互对应的 *n* 列值同时为 1.考虑到 M_3 的最后一列全为

0,M_2 的最后一列全为 1,所以这 a 条和 $N-a$ 条记录均产生于 M 的前 $m-1$ 列中,即这些 n-项集都包含与 M_1 中,假设不成立.

综上所述,满足支持度 *Minsup* 的 n-项集若存在,必在其子矩阵 M_1 或者 M_2 之中.因此,FI-MDC 算法能够正确推导出频繁项集.

3 基于关联规则的多标记分类算法

3.1 多标记学习定义

设 $X=R^d$ 表示样本空间,$Y=\{y_1,y_2,\dots,y_q\}$表示所有可能的类别标记构成的集合,其中,Y 中的任意一项满足 $y_i\in\{0,1\}$,取值为 1 表示该标记为相关标记,反之为无关标记.给定多标记数据集 $DS=\{(X_1,Y_1),(X_2,Y_2),\dots,(X_p,Y_p)\}$,多标记学习框架的目标是训练出一个多标记分类器 $h:X\to 2^Y$.然而在大多数情况下,学习系统的输出往往对应于某个实值函数 $f:X\times Y\to R$,对于给定的训练样本 X_i 及其对应的标记集合 Y_i,一个优秀的学习系统需要在属于 Y_i 的类别标记上较大的值,而在不属于 Y_i 的类别标记上较小的值.也就是说,对于任意的 $y_1\in Y_i$ 以及 $y_2\notin Y_i$,有 $f(x_i,y_1)>f(x_i,y_2)$成立.实值函数 $f(\cdot,\cdot)$还可以转化为一个排序函数 $rank_f(\cdot,\cdot)$,该排序函数将所有的实值输出 $f(x_i,y)(y\in Y)$映射到集合$\{y_1,y_2,\dots,y_q\}$上,使得当$f(x_i,y_1)>f(x_i,y_2)$成立时,$rank_f(x_i,y_1)>rank_f(x_i,y_2)$成立.一般来说,多标记分类器 $h(\cdot)$可以由实数值函数 $f(\cdot,\cdot)$求得,即 $h(x_i)=\{y|f(x_i,y)>t(x_i),y\in Y\}$,其中,$t(\cdot)$为相应的阈值函数,该函数通常设为零常量函数.

3.2 基于全局关联规则挖掘的多标记分类算法

现有的利用关联规则处理多标记分类问题的算法,其主要思想是挖掘出频繁项集并产生关联规则,然后通过一定的方式筛选出部分规则并输出到分类器中.这类算法在一定程度上可以提高多标记算法分类准确率,但本质上仍然是将多标记学习问题转化为多个单标记问题,并且完全忽略标记之间的相关性,造成了标记之间信息的丢失.针对这一问题,我们从标记之间的相关性入手,提出一种基于全局关联规则挖掘的多标记分类算法(GAR-MLC),将 MDC-FIM 算法与传统的多标记分类算法相结合,首先对数据集的所有标记挖掘频繁项集,得到标记之间的关联规则,并对原有训练数据集的标记分布进行更新,最后在更新的数据上采用传统的多标记算法进行训练.这一算法能够充分利用标记内部的相关性,从而达到提高分类准确率的效果.

和已有工作(关联规则为从特征到标记)相比,MDC-FIM 算法得到的关联规则前项和后项都是标记(从标记到标记).MDC-FIM 算法通过调节支持度和置信度,可以得到一系列强关联规则,在 GAR-MLC 算法中,最重要的一步是选择合适的关联规则,并对原有训练数据集的标记分布进行更新,更新的规则是:对满足前项值均为 1(即前项的所有标记均为相关标记)的示例,将规则后项修正为 1(即后项的所有标记置位相关标记).举例说明,假设已经通过 MDC-FIM 得到频繁项集 y_1,y_2,y_3,并求出规则:

$$y_1\wedge y_2\Rightarrow y_3(Confidence=0.85).$$

更新流程是对数据集中所有满足 $y_1=1$ 并且 $y_2=1$ 的示例取出,并依据关联规则将 y_3 的值修正为 1.

GAR-MLC 的主要步骤如下:(1) 对多标记数据集使用 MDC-FIM 算法,找出标记之间满足最小支持度的频繁项集 *FreItem*;(2) 根据第(1)步得到的频繁项集生成关联规则 Rule;(3) 使用第(2)步得到的关联规则对多标记数据进行修改,即根据关联规则全局修改多标记数据集;(4) 采用现有的多标记算法进行训练和预测,最终得到新的分类结果.

算法 **2**. 基于全局关联规则挖掘的多标记分类算法(GAR-MLC).

输入:多标记数据集 *DS*,多标记分类算法 MLC.

输出:新的多标记分类模型 GAR-MLC.

1) 对多标记数据集 *DS* 使用 MDC-FIM 算法求出相应的频繁项集 *FreItem*;
2) 根据频繁项集 *FreItem* 生成关联规则 Rule;
3) 根据关联规则 Rule 修改相应的多标记子数据集 *DS*;
4) 在新的数据集上,使用多标记学习算法 MLC 得出一个新的分类模型 GAR-MLC;

5) 返回新的多标记分类模型 GAR-MLC.

3.3 基于局部关联规则挖掘的多标记分类算法

从本质上看,GAR-MLC 是对多标记数据集使用 MDC-FIM 算法找出满足最小支持度的频繁项集,据此计算出标记之间的关联规则,并基于新的关联规则修正全局多标记数据集,再对新的数据采用传统的多标记算法进行实验,从而达到提高分类准确率的效果.GAR-MLC 中重要的一步是利用关联规则对全局多标记数据集进行修正,假设已经根据 MDC-FIM 算法得出规则 Rule,并且其置信度为β,使用该规则对标记集合修正时,有 1−β的概率修正错误.此外,由于是基于全局数据进行挖掘关联规则,满足一定置信度的规则数也较少.从数据上看,每个数据的特征可以看作超平面上的一个点,标记可以看作超平面的另一个点,多标记分类算法的任务就是寻找从特征到标记的一个映射.类比于单标记学习,一个分类效果好的多标记学习算法应当是在特征构成的超平面上的任意两个点越接近时,其对应标记构造的超平面上的两个点越接近.考虑到直接利用 GAR-MLC 对标记进行修正,忽略了对多标记学习问题中数据特征之间的分析,而且挖掘出的关联规则数较少等问题,我们进一步提出了一种基于局部关联规则挖掘的多标记分类算法(LAR-MLC),首先使用聚类算法对样本进行聚类得到多个簇,并利用 MDC-FIM 算法在每个簇下挖掘对应的关联规则,再使用这些关联规则对相应簇内的数据标记进行修正,修正方法与第 3.2 节的方法一样,最后,在更新后的数据上训练多标记学习分类算法.与 GAR-MLC 相比,LAR-MLC 主要有以下优点:第一,LAR-MLC 在数据集上基于特征先对样本进行聚类,再对簇内标记使用 MDC-FIM 算法,充分考虑到示例特征和标记对分类结果的影响;第二,使用聚类算法时,将特征构成的超平面内接近的点归为一个类别,每个类别对应的标记往往相同或相近,再对每个类别寻找关联规则时,对于相同规则,LAR-MLC 计算得到的置信度大于 GAR-MLC.这样就能得到较多的关联规则,也使得对数据的修正更加合理.

LAR-MLC 的主要步骤如下:(1) 对多标记数据使用聚类算法进行聚类,目的是将多标记数据转化为多个子数据集;(2) 使用 MDC-FIM 算法找出各个子数据集内标记之间的满足最小支持度的频繁项集 *FreItem*;(3) 根据第(2)步得到的频繁项集生成局部关联规则 Rule;(4) 使用第(3)步中得到的局部关联规则对相应的多标记子数据进行修改;(5) 整合所有子数据集,应用相应多标记分类算法得到最终分类结果.

算法 **3**. 基于局部关联规则挖掘的多标记分类算法(LAR-MLC).

输入:多标记数据集 *DS*,多标记算法 MLC.

输出:新的多标记分类模型 LAR-MLC.

1) 对多标记数据集 *DS* 使用聚类算法进行聚类,并把 *DS* 转化为多个子数据集;
2) 对每个子数据集分别使用 MDC-FIM 算法求出相应的频繁项集 *FreItem*;
3) 根据频繁项集 *FreItem* 生成关联规则 Rule;
4) 根据关联规则 Rule 修改相应的多标记子数据集;
5) 在整合后的新数据集上使用多标记学习算法 MLC 得出一个新的分类模型 LAR-MLC;
6) 返回新的多标记分类模型 LAR-MLC.

GAR-MLC 和 LAR-MLC 的本质都是对多标记数据集使用 MDC-FIM 算法找出满足最小支持度的频繁项集,据此计算出标记之间的关联规则,并基于关联规则修正多标记数据集,再对更新后的数据采用传统的多标记算法进行实验,从而达到提高分类准确率的效果.不同点主要在于:GAR-MLC 直接基于全局数据挖掘关联规则,再对多标记数据作全局的修改;而 LAR-MLC 算法首先将多标记数据集聚类,在每个簇内挖掘局部数据的关联规则,再对多标记数据作局部的修改.

4 实验分析

4.1 实验数据

本文在 6 个多标记真实数据集上进行实验比较,这些数据来自多标记的不同领域.Emotions 来自于歌曲分类,CAL-500 来自于音频信号,Flags 来自于图像识别,Genbase 来自于蛋白质分类,Yeast 来自于基因功能检测,

Corel5k 来自于图像标注.表 1 给出了数据集的统计信息,这些数据集都可以从开源项目 Mulan[26]的主页(http://mulan.sourceforge.net/index.html)下载得到.

Table 1 Experimental datasets

表 1 实验数据集

数据集 *DS*	示例数目 *p*	示例维度 *d*	标记数目 *q*	标记基数 *LCard*	标记密度 *LDen*	标记多样性 *LDiv*
Emotions	593	72	6	1.869	0.311	27
CAL-500	502	68	174	26.044	0.150	502
Flags	194	19	7	3.392	0.485	54
Genbase	662	1 186	27	1.252	0.046	32
Yeast	2 417	103	14	4.237	0.303	198
Corel5k	5 000	499	374	3.52	0.009	3 175

- *LCard* 指的是标记基数(label cardinality),即每个样本相关标记的平均个数:

$$LCard(DS)=\frac{1}{p}\sum_{i=1}^{p}\sum_{j=1}^{q}\langle y_i^{(j)}=1\rangle \tag{3}$$

其中,对于任意的谓词π,当π成立时,$\langle\pi\rangle$取值为 1;否则,$\langle\pi\rangle$取值为 0.

- *LDen* 指的是标记密度(label density),即标记基数相当于标记数目的比例:

$$LDen(DS)=LCard(DS)/q \tag{4}$$

- *LDiv* 指的是标记多样性(label diversity),即相关样本标记集合的总数:

$$LDiv(DS)=|\{y|\exists x:(x,y)\in DS\}| \tag{5}$$

4.2 实验配置

基于第 3.1 节中的符号表示,对给定的测试集 $S=\{(X_1,Y_1),(X_2,Y_2),\ldots,(X_s,Y_s)\}$,我们实验中采用如下评价指标来度量多标记学习系统的性能.

1) Hamming loss.该指标考察样本在单个标记上的误分类情况,即隶属于该样本的概念标记未出现在现在标记集合中或不属于该样本的概念标记出现在标记集合中:

$$hloss_s(h)=\frac{1}{s}\sum_{i=1}^{s}\frac{1}{q}|h(X_i)\Delta Y_i| \tag{6}$$

其中,算子Δ用于表示两个集合之间的对称差,$|\cdot|$为返回集合大小.

2) One-error.该指标考察在样本的概念标记排序序列中,序列最前端的标记不属于样本标记的情况:

$$one\text{-}error_s(f)=\frac{1}{s}\sum_{i=1}^{s}\langle\arg\max_{y\in f} f(x_i,y)\notin Y\rangle \tag{7}$$

3) Coverage.该指标考察在样本的概念标记排序序列中,覆盖隶属于样本的所有概念标记所需的搜索深度情况:

$$coverage_s(f)=\frac{1}{s}\sum_{i=1}^{s}\max_{y\in Y_i} rank_f(x_i,y)-1 \tag{8}$$

4) Ranking loss.该指标考察在样本的概念标记排序序列中出现排序错误的情况:

$$rloss_s(f)=\frac{1}{s}\sum_{i=1}^{s}\frac{1}{|Y_i||\overline{Y}_i|}|\{y_1,y_2\}|f(x_i,y_1)\le f(x_i,y_2),(y_1,y_2)\in Y_i\times\overline{Y}_i| \tag{9}$$

其中,$\overline{Y}_i$ 代表 Y_i 在集合 Y 中的补集.

5) Average precision.该指标考察在样本的概念标记排序序列中,排在隶属于该样本的概念标记之前的标记仍属于样本标记集合的情况:

$$avgprec_s(f)=\frac{1}{s}\sum_{i=1}^{s}\frac{1}{|Y_i|}\sum_{y\in Y_i}\frac{|y'|rank_f(x_i,y')\le rank_f(x_i,y),y'\in Y_i|}{rank_f(x_i,y)} \tag{10}$$

对于前 4 种评价指标而言,指标取值越小,则算法性能越优;对于最后一种评价指标而言,指标取值越大,则

算法性能越优.

我们将 GAR-MLC 和 LAR-MLC 应用于传统的多标记学习模型 ML-KNN,Rank-SVM,BP-MLL 中,分别得到对应的新的多标记学习算法 GAR-ML-KNN,LAR-ML-KNN,GAR-Rank-SVM,LAR-Rank-SVM,GAR-BP-MLL,LAR-BP-MLL,通过上文提到的 5 种评价指标来对比新旧模型在多标记分类学习上的性能.对于本文所提算法,在挖掘关联规则时共有两个参数需要调整.经过部分实验可知:当关联规则最小支持度值 *Minsup* 大于 0.1 时,在不降低最小置信度的前提下得到的关联规则数量太少,因此,我们将寻找频繁项集时选用的最小支持度 *Minsup*=0.1,寻找关联规则时选用的 *Minconf*=0.7.LAR-MLC 算法中,聚类使用的是 *k*-Means 算法,聚类个数设置为 5.分类器参数设置如下:在 ML-KNN 算法中设置近邻个数参数 *k*=10,平滑参数设置为 1.对于 Rank-SVM 算法,设置代价参数 *c*=1,同时选择 RBF 核函数.在 BP-MLL 中,神经网络的隐含神经元个数为特征个数的 20%,学习率 α=0.05 以及训练时最大的迭代次数为 100.

4.3 实验结果与分析

对于每个数据集,本文随机选取 80%的数据为训练集,余下的 20%的数据为测试集,重复实验 10 次.表 2~表 7 显示的是不同数据集在 9 种不同算法下实验结果的均值与标准差.

Table 2 Comparison results of different multi-label algorithms on the Emotions dataset (mean±std)

表 2 多标记算法在 Emotions 数据集上的比较(均值±标准差)

Method	Hamming loss↓	One-Error↓	Coverage↓	Ranking loss↓	Average precision↑
ML-KNN	0.269±0.019	0.378±0.052	2.271±0.199	0.254±0.034	0.718±0.030
GAR-ML-KNN	0.251±0.020	0.327±0.079	2.255±0.186	0.246±0.046	0.734±0.045
LAR-ML-KNN	0.250±0.013	0.367±0.048	2.210±0.161	0.252±0.030	0.736±0.029
Rank-SVM	0.238±0.021	0.362±0.067	2.075±0.236	0.225±0.041	0.739±0.039
GAR-Rank-SVM	0.226±0.021	0.274±0.068	2.063±0.173	0.199±0.035	**0.777±0.036**
LAR-Rank-SVM	**0.224±0.019**	**0.271±0.045**	**1.985±0.176**	**0.190±0.020**	0.765±0.022
BP-MLL	0.314±0.011	0.620±0.078	3.272±0.177	0.457±0.039	0.539±0.035
GAR-BP-MLL	0.333±0.044	0.603±0.146	3.231±0.207	0.440±0.060	0.554±0.054
LAR-BP-MLL	0.342±0.029	0.557±0.063	3.044±0.221	0.415±0.023	0.572±0.018

Table 3 Comparison results of different multi-label algorithms on the CAL-500 dataset (mean±std)

表 3 多标记算法在 CAL-500 数据集上的比较(均值±标准差)

Method	Hamming loss↓	One-Error↓	Coverage(×10)↓	Ranking loss↓	Average precision↑
ML-KNN	0.140±0.006	0.108±0.031	**12.910±0.451**	0.182±0.007	0.494±0.010
GAR-ML-KNN	0.139±0.004	0.108±0.040	13.005±0.351	**0.180±0.009**	**0.510±0.018**
LAR-ML-KNN	0.138±0.004	0.112±0.034	12.990±0.303	0.181±0.005	0.507±0.013
Rank-SVM	0.189±0.024	0.230±0.057	13.713±0.355	0.247±0.023	0.439±0.022
GAR-Rank-SVM	0.188±0.026	0.216±0.048	13.588±0.362	0.243±0.021	0.442±0.014
LAR-Rank-SVM	0.189±0.027	0.216±0.044	13.561±0.345	0.241±0.022	0.444±0.027
BP-MLL	0.138±0.004	0.132±0.048	13.043±0.214	0.191±0.003	0.482±0.006
GAR-BP-MLL	0.136±0.003	0.124±0.040	13.198±0.439	0.189±0.007	0.501±0.007
LAR-BP-MLL	**0.131±0.006**	**0.094±0.040**	12.971±0.378	0.181±0.007	0.502±0.014

Table 4 Comparison results of different multi-label algorithms on the Flags dataset (mean±std)

表 4 多标记算法在 Flags 数据集上的比较(均值±标准差)

Method	Hamming loss↓	One-Error↓	Coverage↓	Ranking loss↓	Average precision↑
ML-KNN	0.330±0.034	0.232±0.086	4.069±0.333	0.264±0.053	0.783±0.048
GAR-ML-KNN	0.278±0.036	0.227±0.095	**3.742±0.305**	**0.201±0.046**	0.823±0.032
LAR-ML-KNN	0.264±0.041	0.231±0.073	3.992±0.243	0.215±0.035	0.812±0.030
Rank-SVM	0.286±0.060	0.246±0.147	4.042±0.311	0.232±0.068	0.807±0.074
GAR-Rank-SVM	0.259±0.036	0.210±0.121	3.821±0.302	0.226±0.047	0.813±0.047
LAR-Rank-SVM	**0.255±0.038**	**0.207±0.060**	3.815±0.370	0.209±0.019	0.821±0.036
BP-MLL	0.281±0.033	0.222±0.080	4.000±0.258	0.233±0.044	0.803±0.036
GAR-BP-MLL	0.265±0.045	0.218±0.084	3.932±0.354	0.215±0.051	0.821±0.036
LAR-BP-MLL	0.260±0.026	0.212±0.072	3.825±0.296	0.211±0.042	**0.825±0.035**

Table 5 Comparison results of different multi-label algorithms on the Genbase dataset (mean±std)

表 5 多标记算法在 Genbase 数据集上的比较(均值±标准差)

Method	Hamming loss(×0.1)↓	One-Error↓	Coverage↓	Ranking loss(×0.1)↓	Average precision↑
ML-KNN	0.050±0.019	0.008±0.001	0.533±0.399	0.053±0.075	0.990±0.010
GAR-ML-KNN	0.057±0.027	0.009±0.001	0.502±0.269	0.039±0.044	0.992±0.010
LAR-ML-KNN	0.138±0.004	0.006±0.001	0.496±0.246	0.037±0.040	0.986±0.010
Rank-SVM	0.011±0.007	0.006±0.001	0.282±0.101	0.010±0.025	0.996±0.005
GAR-Rank-SVM	0.008±0.005	0.006±0.001	0.283±0.128	0.009±0.002	0.998±0.003
LAR-Rank-SVM	**0.007±0.005**	**0.005±0.001**	**0.279±0.128**	**0.008±0.002**	**0.999±0.003**
BP-MLL	0.131±0.042	0.009±0.001	0.764±0.168	0.138±0.036	0.893±0.028
GAR-BP-MLL	0.118±0.027	0.008±0.001	0.533±0.192	0.101±0.031	0.899±0.038
LAR-BP-MLL	0.051±0.029	0.007±0.001	0.431±0.298	0.042±0.026	0.903±0.006

Table 6 Comparison results of different multi-label algorithms on the Yeast dataset (mean±std)

表 6 多标记算法在 Yeast 数据集上的比较(均值±标准差)

Method	Hamming loss↓	One-Error↓	Coverage↓	Ranking loss↓	Average precision↑
ML-KNN	0.199±0.012	0.242±0.044	6.309±0.206	0.170±0.016	0.759±0.027
GAR-ML-KNN	0.194±0.007	0.233±0.009	6.218±0.152	0.167±0.007	0.764±0.007
LAR-ML-KNN	0.191±0.009	0.226±0.029	**6.202±0.218**	0.163±0.013	0.770±0.018
Rank-SVM	0.190±0.010	0.230±0.020	6.298±0.294	0.167±0.014	0.770±0.015
GAR-Rank-SVM	**0.187±0.008**	0.222±0.028	6.591±0.156	0.150±0.011	0.808±0.014
LAR-Rank-SVM	0.190±0.008	**0.217±0.021**	6.317±0.236	**0.147±0.012**	**0.812±0.016**
BP-MLL	0.217±0.091	0.244±0.027	6.596±0.205	0.180±0.011	0.746±0.016
GAR-BP-MLL	0.216±0.060	0.241±0.026	6.545±0.228	0.179±0.012	0.749±0.016
LAR-BP-MLL	0.211±0.010	0.237±0.019	6.511±0.220	0.176±0.011	0.750±0.013

Table 7 Comparison results of different multi-label algorithms on the Corel5k dataset (mean±std)

表 7 多标记算法在 Corel5k 数据集上的比较(均值±标准差)

Method	Hamming loss↓	One-Error↓	Coverage↓	Ranking loss↓	Average precision↑
ML-KNN	0.014±0.000	0.587±0.022	127.14±3.70	0.208±0.008	0.354±0.016
GAR-ML-KNN	0.013±0.000	0.584±0.020	123.34±3.19	0.201±0.007	0.351±0.012
LAR-ML-KNN	0.013±0.000	0.579±0.019	121.74±5.18	0.198±0.007	0.359±0.015
Rank-SVM	**0.009±0.000**	0.569±0.016	117.22±5.19	0.162±0.008	0.357±0.009
GAR-Rank-SVM	**0.009±0.000**	**0.563±0.019**	**107.72±4.97**	**0.154±0.009**	0.359±0.009
LAR-Rank-SVM	**0.009±0.000**	0.568±0.020	109.72±5.12	0.158±0.006	**0.362±0.009**
BP-MLL	0.014±0.001	0.597±0.027	143.4±0.29	0.182±0.005	0.325±0.016
GAR-BP-MLL	0.013±0.000	0.579±0.027	140.9±0.27	0.162±0.005	0.329±0.009
LAR-BP-MLL	0.013±0.000	0.582±0.027	128.4±0.24	0.166±0.006	0.331±0.010

由表 2~表 7 给出的 9 种多标记分类算法在 5 个评价指标上的实验结果可以看出:相对于现有的 ML-KNN, Rank-SVM,BP-MLL 等多标记算法,本文提出的 GAR-MLC 和 LAR-MLC 算法对多标记数据集修正是有效的, 对于对应的多标记分类算法都能在相应评价指标上有所提升.对于 LAR-MLC 和 GAR-MLC 之间的比较,在 6 个数据集上的大部分评价指标上,LAR-MLC 的表现要优于 GAR-MLC.

综合比较 6 个数据集可以看出,Rank-SVM 及其改进模型 GAR-Rank-SVM,LAR-Rank-SVM 在 Emotions, Genbase,Yeast,Corel5k 这 4 个数据集中,在 5 个评价指标上的效果基本都优于其他算法,在其他数据集上表现也较优.为了验证 GAR-MLC 和 LAR-MLC 算法在不同比例训练样本下的稳定性,我们采用 Rank-SVM 及其改进模型 GAR-Rank-SVM,LAR-Rank-SVM 对相同数据集上进行实验,实验共分 3 组,训练集的数据占总数据的比例分别为 40%,60%,80%.以 Yeast 数据集为例,表 8 给出在不同训练集比例下,3 种算法在 5 种评价指标下的分类效果.

从表 8 中可以看出,在训练集不同的数目下(40%,60%,80%),GAR-MLC,LAR-MLC 和 Rank-SVM 算法相比, 在不同评价指标上都有一定程度的提升.这说明 GAR-MLC,LAR-MLC 算法是稳定的.另一方面,随着训练集样本的增加,对应的多标记算法分类效果也会有一定程度的提高.此外,我们同样做了其他 5 个数据集上的稳定性对比实验,其比较结果和表 8 一致,限于篇幅所限,在此不再给出.

Table 8 Comparison results of Rank-SVM, GAR-Rank-SVM and LAR-Rank-SVM on the Yeast dataset (mean±std)

表 **8** Rank-SVM,GAR-Rank-SVM 和 LAR-Rank-SVM 算法在 Yeast 数据集上的比较(均值±标准差)

Method	Hamming loss↓	One-Error↓	Coverage↓	Ranking loss↓	Average precision↑	Percentage
Rank-SVM	0.203±0.003	0.244±0.011	6.575±0.073	0.174±0.004	0.756±0.006	40%
GAR-Rank-SVM	0.202±0.002	0.247±0.010	6.528±0.081	0.179±0.003	0.755±0.004	40%
LAR-Rank-SVM	0.202±0.003	0.243±0.008	6.525±0.073	0.178±0.005	0.756±0.005	40%
Rank-SVM	0.199±0.005	0.239±0.023	6.447±0.122	0.176±0.009	0.759±0.012	60%
GAR-Rank-SVM	0.198±0.005	0.238±0.023	6.417±0.089	0.172±0.008	0.764±0.014	60%
LAR-Rank-SVM	0.195±0.004	0.227±0.018	6.416±0.133	0.169±0.006	0.768±0.009	60%
Rank-SVM	0.190±0.010	0.230±0.020	**6.298±0.294**	0.167±0.014	0.770±0.015	80%
GAR-Rank-SVM	**0.187±0.008**	0.222±0.028	6.591±0.156	0.150±0.011	0.808±0.014	80%
LAR-Rank-SVM	0.190±0.008	**0.217±0.021**	6.317±0.236	**0.147±0.012**	**0.812±0.016**	80%

对于 MDC-FIM 算法挖掘出的关联规则是否有效,我们也基于不同规模数据集进行了分析.对于只具有少量标记的数据集或标记基数较少的数据集,以 Emotions 数据集为例,该数据来源于歌曲情感分类,数据集共有 6 个标记,且标记基数仅为 1.869.根据 *Minsup*=0.1 和 *Minconf*=0.7,我们能够得到 2 条关联规则,分别为“Relaxing-Clam⇒Quiet-Still”和“Sad-lonely⇒Quiet-Still”.从语义上分析,一首“放松”的歌或者“悲伤”的歌,其曲调通常都是舒缓的,易被归类为“安静”的歌曲,这与我们对歌曲情感的认知是一致的.而对于标记数量较大或标记基数较大的数据集,以 CAL-500 数据集为例,数据集共有 174 个标记,标记基数为 26.044,我们可以得到 301 条规则,将所有的规则用于修改数据集,会削弱算法所使用的多类标分类器在最终分类结果中的作用,而过于倚重关联规则的分类性能.为此,我们可以选用置信度排序前 10%的规则用于修改数据集,这样既在一定程度上保留了标记之间的相关性,也能避免过拟合现象.

MDC-FIM 算法在挖掘频繁项集时共有两个参数需要调整,即最小支持度 *Minsup* 和最小置信度 *Minconf*.在我们之前的实验中,这两个参数的取值分别为 *Minsup*=0.1 和 *Minconf*=0.7.为了分析 MDC-FIM 算法在不同参数设置下挖掘出的关联规则的数目的变化规律,我们对这两个参数进行敏感性分析.图 3 给出了 GAR-Rank-SVM 和 LAR-Rank-SVM 在 Yeast 数据集上,关联规则的数目指标随 *Minsup* 和 *Minconf* 的变化影响.

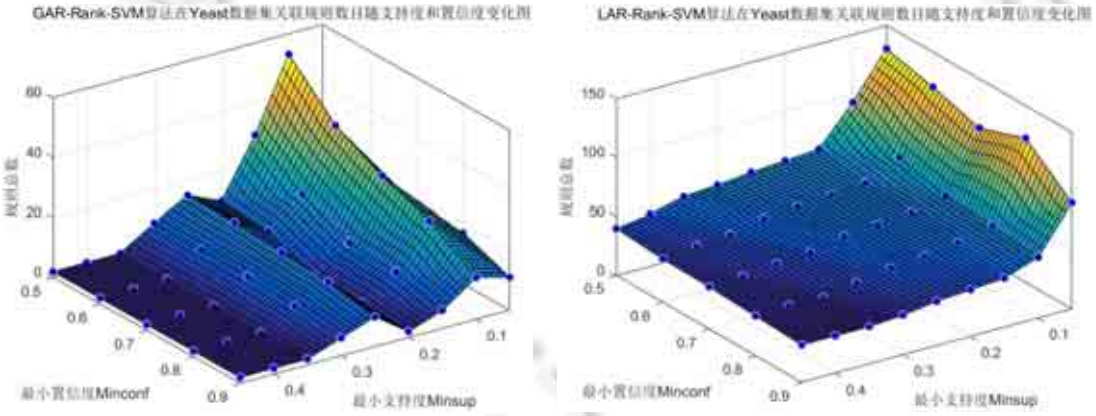

Fig.3 Number of association rules of GAR-Rank-SVM and LAR-Rank-SVM on Yeast dataset varies with *Minsup* and *Minconf*

图 3 GAR-Rank-SVM 和 LAR-Rank-SVM 在 Yeast 数据集上的关联规则数目指标随 *Minsup* 和 *Minconf* 的变化

从图 3 中可以看出:当固定 *Minsup* 时,挖掘出的关联规则的数目随 *Minconf* 的增大而减小;当固定 *Minconf* 时,挖掘出的关联规则数目随 *Minsup* 的增大而减小.LAR-Rank-SVM 算法将原始数据集分为多个类别,并在每个类别中寻找关联规则,因此,得到的关联规则数目比 GAR-Rank-SVM 多.当 *Minsup* 大于 0.2 或 *Minconf* 大于

0.8 时,得到的关联规则数量较少.接下来,我们检验依据参数选取出的关联规则数量与分类准确率之间的变化情况.从图 3 可以看出:当 *Minsup* 取值在 0.1、*Minconf* 取值为 0.7 附近时,选取出的规则数目较多.对此,我们检验在这两个参数值附近时分类性能的变化.图 4 给出 GAR-Rank-SVM 和 LAR-Rank-SVM 在 Yeast 数据集上,Average Precision 指标随 *Minconf* 和 *Minconf* 的变化影响.

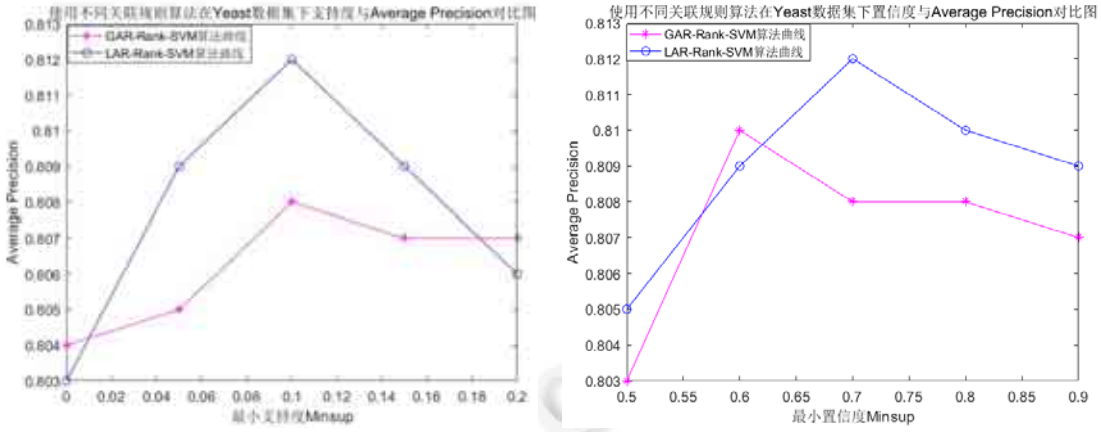

Fig.4 Average precision of GAR-Rank-SVM and LAR-Rank-SVM on Yeast dataset varies with *Minsup* and *Minconf*

图 4 GAR-Rank-SVM 和 LAR-Rank-SVM 在 Yeast 数据集上的 AP 指标随 *Minsup* 和 *Minconf* 的变化

从图 4 中可以看出:当 *Minsup* 取 0.1 和 *Minconf* 取 0.7 时,GAR-Rank-SVM 在 Yeast 数据集上能够取得最大的分类准确率;当 *Minsup* 取 0.1 和 *Minconf* 取 0.6 时,LAR-Rank-SVM 能够取得最大的分类准确率;而在 *Minconf* 取 0.7 时,能够取得次优解.

在实验的 6 个数据集中,CAL-500 数据集的标记基数最大,这意味着在 *Minsup* 和 *Minconf* 选取同等数值情况下,得到的关联规则会更多,从而带来的时间消耗也会更大.为了检验 MDC-FIM 算法在大规模数据标记集合上的运行效率,我们在该数据集上设计了对比实验,将 MDC-FIM 算法与其他经典关联规则对比在计算频繁项集时在时间上的消耗,对比算法为 Apriori 算法、改进的 Apriori 算法、Eclat 算法,最小置信度 *Minconf* 设置为 0.7.各算法在不同支持度计算频繁项集所消耗的时间如图 5 所示.

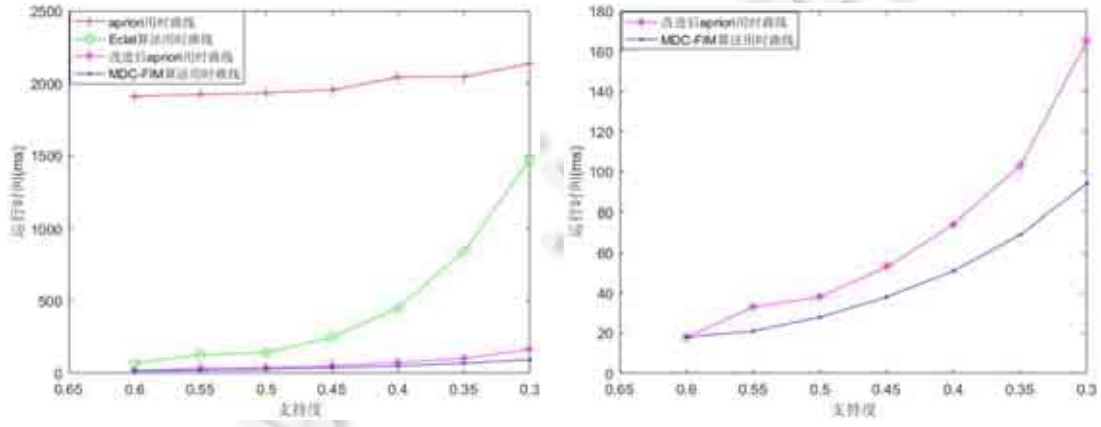

Fig.5 Average running time of different association rule algorithms on CAL-500 dataset

图 5 不同关联规则算法在 CAL-500 数据集上的平均运行时间

从图 5 可以看出:随着支持度的减小,各算法所消耗的时间逐渐增加;与其他算法相比,本文提出的 MDC-FIM 算法挖掘频繁项集在时间效率上要明显占优.

通过上述 4 组不同实验结果可以看出:MDC-FIM 算法能够快速而准确地找到标记之间的关联规则,将 GAR- MLC 和 LAR-MLC 应用于现有的 ML-KNN,Rank-SVM,BP-MLL 多标记算法,得到对应的改进多标记学习算法分类效果都要优于原先的算法;并且大部分情况下,LAR-MLC 效果要优于 GAR-MLC.通过对示例数量和标记数量进行分析可以看出,MDC-FIM 可以适用于各种多标记数据集.

5 总 结

本文提出了一种基于矩阵分治的频繁项集挖掘算法,并将其应用于多标记学习框架中,分别提出了基于全局关联规则挖掘和局部关联规则挖掘的多标记分类算法.这两种算法通过挖掘标记之间的关联规则,从全局和局部两个角度出发对原有数据进行更新;在此基础上,可以直接应用现有的多标记学习算法得到新的多标记分类模型.多个真实的数据集上的实验结果表明:所提出的算法适用于多标记学习框架,且能够取得良好的分类性能.当前工作主要把标记之间的关联性应用于数据更新,在未来的工作中,我们将结合所提出的算法对多标记分类算法做进一步的改进,以期获得更好的分类效果.

References:

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

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

[3] Rubin TN, Chambers A, Smyth P, Steyvers M. Statistical topic models for multi-label document classification. Machine Learning, 2012,88(1):157−208. [doi: 10.1007/s10994-011-5272-5]

[4] Cabral RS, Torre FDL, Costeira JP, Bernardino A. Matrix completion for multi-label image classification. In: Proc. of the Advances in Neural Information Processing Systems. 2011. 190−198.

[5] Zhou Y, Xue H, Geng X. Emotion distribution recognition from facial expressions. In: Proc. of the ACM Int'l Conf. on Multimedia. 2015. 1247−1250. [doi: 10.1145/2733373.2806328]

[6] Lo HY, Wang JC, Wang HM, Lin SD. Cost-Sensitive multi-label learning for audio tag annotation and retrieval. IEEE Trans. on Multimedia, 2011,13(3):518−529. [doi: 10.1109/TMM.2011.2129498]

[7] Wang J, Zhao Y, Wu X, Hua XS. A transductive multi-label learning approach for video concept detection. Pattern Recognition, 2011,44(10-11):2274−2286. [doi: 10.1016/j.patcog.2010.07.015]

[8] Gibaja E, Ventura S. A tutorial on multilabel learning. ACM Computing Surveys, 2015,47(3):1−38. [doi: 10.1145/2716262]

[9] Liu B, Hsu W, Ma Y. Integrating classification and association rule mining. In: Proc. of the Knowledge Discovery and Data Mining (KDD). 1970. 80−86.

[10] Li W, Han J, Pei K. CMAR: Accurate and efficient classification based on multiple class-association rules. In: Proc. of the Int'l Conf. on Data Mining. 2001. 369−376. [doi: 10.1109/ICDM.2001.989541]

[11] Yin X, Han J. CPAR: Classification based on predictive association rules. In: Proc. of the Int'l Conf. on Data Mining. 2003. 35−42.

[12] Spyromitros E, Tsoumakas G, Vlahavas I. An empirical study of lazy multilabel classification algorithms. In: Proc. of the Artificial Intelligence: Theories, Models and Applications. 2008. 401−406. [doi: 10.1007/978-3-540-87881-0_40]

[13] 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]

[14] Zhang ML. An improved multi-label lazy learning approach. Journal of Computer Research and Development, 2012,49(11): 2271−2282 (in Chinese with English abstract).

[15] Elisseeff AE, Weston J. A kernel method for multi-labelled classification. In: Proc. of the Advances in Neural Information Processing Systems. 2002. 681−687.

[16] Zhang ML, Zhou ZH. Multilabel neural networks with applications to functional genomics and text categorization. IEEE Trans. on Knowledge and Data Engineering, 2006,18(10):1338−1351. [doi: 10.1109/TKDE.2006.162]

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

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

[19] 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.

[20] Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules: Readings in Database Systems. 3rd ed., Morgan Kaufmann Publishers Inc., 1998. 580−592.

[21] Nguyen TL, Vo B, Snasel V. Efficient algorithms for mining colossal patterns in high dimensional databases. In: Proc. of the Knowledge-Based Systems. 2017. 75−89. [doi: 10.1016/j.knosys.2017.01.034]

[22] Patel K, Kapadia N, Parikh M. Discover multi-label classification using association rule mining. Int'l Journal of Advance Engineering and Research Development, 2014,1(1):18−23. [doi: 10.21090/ijaerd.0101004]

[23] Alazaidah R, Thabtah F, Al-Radaideh QA. A multi-label classification approach based on correlations among labels. Int'l Journal of Advanced Computer Science and Applications, 2015,6(2):52−59. [doi: 10.14569/IJACSA.2015.060208]

[24] Li B, Li H, Wu M, Li P. Multi-Label classification based on association rules with application to scene classification. In: Proc. of the Int'l Conf. for Young Computer Scientists. 2008. 36−41. [doi: 10.1109/ICYCS.2008.524]

[25] Charte F, Rivera A, Jesus MJ, Herrera F. Improving multi-label classifiers via label reduction with association rules. In: Proc. of the Int'l Conf. on Hybrid Artificial Intelligent Systems. 2012. 188−199. [doi: 10.1007/978-3-642-28931-6_18]

[26] Tsoumakas G, Vilcek J, Xioufits ES. Mulan: A Java library for multi-label learning. 2011. http://mulan.sourceforge.net/datasets.html

附中文参考文献:

[14] 张敏灵.一种新型多标记懒惰学习算法.计算机研究与发展,2012,49(11):2271−2282.

刘军煜(1993－),男,江苏南通人,硕士生,主要研究领域为机器学习,数据挖掘.

贾修一(1983－),男,博士,副教授,CCF 高级会员,主要研究领域为机器学习,粒计算,数据挖掘.