**基于WEKA的数据挖掘算法学习（九）**

# 多标签分类介绍

上周学习了多标签分类算法，这周接着上周继续就多标签分类进行学习。

每个样本属于一个L样本集合的一个子集，多标签分类(multilabel classification)分类器会给一个新的实例指定多个类别。分类模型有很广泛的实际应用，如：一个文档可能同时属于多个分类；一个蛋白质可能具有多个功能。

# 多标签分类算法介绍

## 2.1 BR

BR二元关系法，为每一个标签产生一个单独的数据集，总共产生n个数据集(j=1…n)，每个数据集包含训练集中的全部样本。对于数据集，包含在该数据集中的所有样本仅被标注是否属于，用该数据集训练一个仅输出样本是否属于的二分类器，为每一个标签训练一个二分类器。对于一个未知类别的样本，将全部二分类器结果组合在一起即得到该样本最终的标签分类结果。

## 2.2 ECC

ECC组合分类器链是对于BR方法的一种改进，针对BR未能考虑标签之间的联系,而导致一定程度上的信息损失,采用了CC(ClassfierChain)分类器链，即通过将BR产生的ｎ 个二分类器连接成一条链，训练样本每经过一个二分类器， 就将其预测结果添至样本属性向量继续代入下一个二分类器中训练。但是由于CC中二分类器不同顺序对结果有较大影响，因此ECC采取多条随机产生的不同标签序列的CC组合,以减轻单个CC由内部二分类器排列顺序问题而带来的不利影响。

## 2.3 MLKNN

改进的kNN算法来处理多标签数据法，首先要找出待预测样本的ｋ个最近邻样本，但是它们在样本标签集的聚合方面采用了不同的方法。其中较为典型的是MLkNN，该方法基于待预测样本的ｋ个最近邻类别标签出现的先验和后验概率，使用最大化后验原则确定待 预测样本的标签集。

## 2.4 HOMER

通过对机器学习领域的很多研究，发现数据集关于类别的分布往往表现出偏斜(skewed)或称不均衡(imbalanced)即类别间样本数量可能存在数量级的差距,这是导致分类效果很不理想的一个重要因素。针对数据偏斜问题，提出了一种层次化结构方法HOMER，使用一种均匀的k-means聚 类方式将整个数据集聚成大小相近的几个类，然后在每个类中训练多标签分类器，并在其上层再构建一个多标签分类器，由此可以根据这样的两层多标签分类模式，使数据集能够根据样本之间的相似度聚类产生近似均匀分布，避免数据偏斜。该算法对标签集Ｌ所包含的元素个数具有线性的训练复杂度和对数的测试复杂度。该方法可以通过均匀的聚类保证上层的多标签分类器处理均匀的类别，但是下层通过非监督的聚类方式产生的每个 类别不能保证该类别下的样本分布也是均匀的，所以数据偏斜问题在HOMER中并未得到完全解决。

# 3.weka算法实现

## 3.1数据集的准备

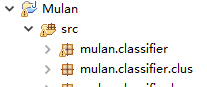
1、MULAN多标签数据集：yeast

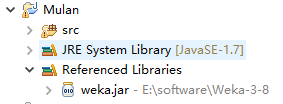
2、MULAN多标签数据集：medical

## 3.2 算法实现

### 3.2.1 MyEclipse中运行Mulan

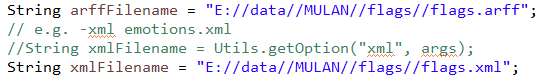
1、由于WEKA不能处理多标签数据集，上周使用MEKA GUI进行了多标签数据集。本次实验使用了在Eclipse中运行Mulan。

2、首先下载安装包，解压，将mulan目录下src文件加入到java项目中：。

3、右击java工程 Properties Libraries选项卡，单击Add External JARs，添加weka安装目录下的weka.jar: 。

4、在所建java项目中找到并打开交叉验证：，

5、然后修改arrf文件路径：



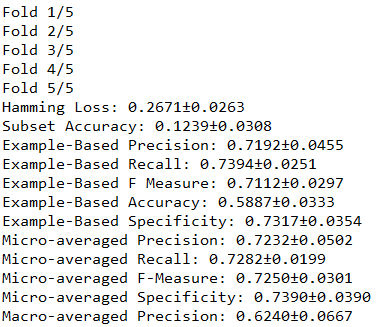
6、修改默认10折交叉验证为5折交叉验证：



7、选择测试多标签算法为MLKNN或其他算法：



8、经过一折一折（设置5折交叉验证）交叉验证，运行得到数据测试结果：



9、由以上过程分别在MyEclipse中测试BR、ECC、MLKNN、HOMER四种算法的测试，记录实验结果。

### 3.2.2 BR

**1、首先选择多标签数据集yeast**

选择BR算法，选择五折交叉验证，分别选择J48和选择NaiveBayes作为为基分类器进行数据集测试：



结果如下：

|  |  |  |
| --- | --- | --- |
|  | yeast数据集 | |
| J48基分类器 | NaiveBayes基分类器 |
| Hamming Loss | 0.2532±0.0358 | 0.3015±0.0069 |
| coverage | 4.0408±0.1325 | 7.4907±0.0675 |
| one-error | 0.3401±0.0779 | 0.3517±0.0199 |
| Ranking loss | 0.2717±0.0345 | 0.2571±0.0047 |
| Avg Precision | 0.7763±0.0330 | 0.6654±0.0101 |
| time | 45193毫秒 | 11396毫秒 |

对于yeast数据集，Avg Precision值在J48基分类器下的BR小于NaiveBayes基分类器下的BR算法的结果，其余指标除Avg Precision均低于NavieBayes基分类器，可见在J48分类器下，BR性能更好。但从时间效率上来说，NaiveBayes优于J48。

**2、再选择多标签数据集medical：**

选择BR算法，选择五折交叉验证，分别选择J48和选择NaiveBayes作为基分类器进行数据集测试，结果如下：

|  |  |  |
| --- | --- | --- |
|  | medical数据集 | |
| J48基分类器 | NaiveBayes基分类器 |
| Hamming Loss | 0.0101±0.0012 | 0.0245±0.0014 |
| coverage | 4.0976±0.8325 | 7.4364±0.5277 |
| one-error | 0.1821±0.0272 | 0.4508±0.0360 |
| Ranking loss | 0.0697±0.0158 | 0.1410±0.0090 |
| Avg Precision | 0.8394±0.0260 | 0.6075±0.0281 |
| time | 113021毫秒 | 70167毫秒 |

对于medical数据集，Avg Precision值在J48基分类器下的BR大于NaiveBayes基分类器下的BR算法的结果，其余指标均低于NavieBayes基分类器，可见在J48分类器下，BR性能更好。与yeast数据集相同，从时间效率上来说， J48时间效率欠佳。

### 2.2.3 ECC

**1、首先选择多标签数据集yeast：**

选择ECC算法，选择五折交叉验证，分别选择J48和选择NaiveBayes作为为基分类器，并设置aNumOfModels=5进行数据集测试，结果如下：

|  |  |  |
| --- | --- | --- |
|  | yeast数据集 | |
| J48基分类器 | NaiveBayes基分类器 |
| Hamming Loss | 0.2253±0.0080 | 0.3015±0.0069 |
| coverage | 7.0740±0.1842 | 7.4907±0.0675 |
| one-error | 0.2909±0.0128 | 0.3517±0.0199 |
| Ranking loss | 0.2132±0.0121 | 0.2571±0.0047 |
| Avg Precision | 0.7126±0.0139 | 0.6654±0.0101 |
| time | 238408毫秒 | 11396毫秒 |

对于yeast数据集，在ECC算法下，同样在J48比在NaiveBayes基分类器下性能更佳，时间效率差些。相比在3.2.2节中yeast数据集ER算法下的结果，ECC下算法性能明显提高。

**2、再选择多标签数据集medical：**

选择ECC算法，选择五折交叉验证，分别选择J48和选择NaiveBayes作为为基分类器，并设置aNumOfModels=5进行数据集测试，结果如下：

|  |  |  |
| --- | --- | --- |
|  | medical数据集 | |
| J48基分类器 | NaiveBayes基分类器 |
| Hamming Loss | 0.0101±0.0012 | 0.2904±0.0059 |
| coverage | 4.0976±0.8325 | 7.5201±0.0843 |
| one-error | 0.1821±0.0272 | 0.3293±0.0300 |
| Ranking loss | 0.0697±0.0158 | 0.2561±0.0071 |
| Avg Precision | 0.8394±0.0260 | 0.6748±0.0112 |
| time | 113021毫秒 | 46044毫秒 |

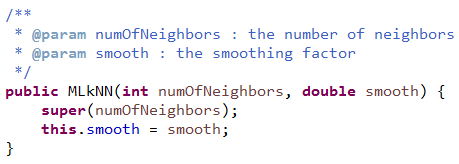
这里可以看见，在J48分类器下，效果仍然要好很多。

通过BR多标签分类算法和ECC多标签分类算法跑出来的结果对比来，ECC算法性能确实优于BR分类算法， ECC针对BR未能考虑标签之间的联系,而导致一定程度上的信息损失,采用了CC(ClassfierChain)分类器链。这里ECC参数aNumOfModels为5，还可以设置其他数值，对算法针对某个数据集性能将会有改变。

### 2.2.4 MLKNN

**1、首先选择多标签数据集yeast：**

选择MLKNN算法，选择五折交叉验证，这里MLKNN有两个参数，the number of neighbors 即K值，the smoothing factor 即平滑因子（默认为1.0），通过调整这两个参数，测试数据，



由于之前跑KNN算法时，得出经验是K值小时模型复杂，建模时间长，但是效率会明显增加，通常越小性能越好，这里选择K值为1、3时进行测试。

结果如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | K=1  S=0.5 | K=1  S=1.0 | K=1  S=1.5 | K=3  S=0.5 | K=3  S=1.0 | K=3  S=1.5 |
| Hamming Loss | 0.2112±0.0108 | 0.2112±0.0108 | 0.2112±0.0108 | 0.1983±0.0097 | 0.1983±0.0097 | 0.1983±0.0097 |
| coverage | 6.3185±0.1721 | 6.3235±0.1718 | 6.3235±0.1718 | 6.2992±0.1346 | 6.3000±0.1317 | 6.3000±0.1315 |
| one-error | 0.2499±0.0243 | 0.2499±0.0243 | 0.2499±0.0243 | 0.2458±0.0145 | 0.2458±0.0145 | 0.2449±0.0144 |
| Ranking loss | 0.1827±0.0107 | 0.1827±0.0107 | 0.1827±0.0107 | 0.1726±0.0104 | 0.1726±0.0103 | 0.1726±0.0103 |
| Avg Precision | 0.7415±0.0159 | 0.7414±0.0159 | 0.7414±0.0159 | 0.7569±0.0145 | 0.7569±0.0143 | 0.7567±0.0142 |
| time | 65455毫秒 | 62948毫秒 | 55902毫秒 | 60132毫秒 | 65125毫秒 | 62592毫秒 |

对于yeast数据集，当K值为1时性能比K值为3时差；K值为1时，增大平滑因子对结果没什么影响，但平滑因子变大，时间效率提高。K值为3时，改变平滑因子对性能有微调，时间效率上改变不大。

**2、首先选择多标签数据集medical：**

选择MLKNN算法，选择五折交叉验证， 改变K值和S值进行数据测试：

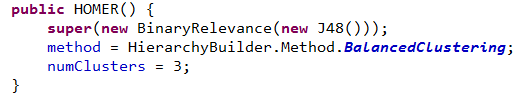
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | K=1  S=0.5 | K=1  S=1.0 | K=1  S=1.5 | K=3  S=0.5 | K=3  S=1.0 | K=3  S=1.5 |
| Hamming Loss | 0.0178±0.0006 | 0.0178±0.0006 | 0.0178±0.0006 | 0.0162±0.0014 | 0.0162±0.0014 | 0.0162±0.0013 |
| coverage | 4.4984±0.4062 | 4.5005±0.4088 | 4.5055±0.4037 | 3.2197±0.2707 | 3.2361±0.2527 | 3.2463±0.2458 |
| one-error | 0.3006±0.0094 | 0.2986±0.0087 | 0.2986±0.0087 | 0.2802±0.0373 | 0.2791±0.0358 | 0.2802±0.0343 |
| Ranking loss | 0.0777±0.0071 | 0.0777±0.0070 | 0.0778±0.0069 | 0.0523±0.0063 | 0.0524±0.0062 | 0.0527±0.0061 |
| Avg Precision | 0.7244±0.0090 | 0.7255±0.0092 | 0.7254±0.0091 | 0.7801±0.0223 | 0.7801±0.0222 | 0.7796±0.0217 |
| time | 8337毫秒 | 8821毫秒 | 8271毫秒 | 7615毫秒 | 8865毫秒 | 8241毫秒 |

对于medical数据集，当K值为1时性能比K值为3时差；改变平滑因子效果不明显。

### 2.2.5 HOMER

**选择多标签数据集yeast、medical：**

选择HOMER算法，选择五折交叉验证，这里HOMER有三个参数：MultilabelLearner、clusters、method。MultilabelLearner在之前BR算法中，yeast和medical均为结果为J48更好，因此这里选择默认值：



结果如下：

|  |  |  |
| --- | --- | --- |
| 数据集选择 | Yeast数据集 | Medical数据集 |
| Hamming Loss | 0.2598±0.0107 | 0.0120±0.0013 |
| coverage | 9.4770±0.1207 | 6.4737±0.9047 |
| one-error | 0.3157±0.0596 | 0.2373±0.0308 |
| Ranking loss | 0.3398±0.0198 | 0.1152±0.0176 |
| Avg Precision | 0.6098±0.0229 | 0.7744±0.0243 |
| time | 58189毫秒 | 118262毫秒 |

Medical数据集时间耗费差不多是yeast数据集的两倍，但样本却是100倍左右，这是由于yeast标签太多（103）。性能上medical跑出的结果更佳优越。

# 4.本周学习心得体会

本周继续学习了多标签分类算法的数据测试，上周进行了meka学习与使用，本次测试使用了在Myeclipse中运行Mulan进行了BR、ECC、MLKNN、HOMER四个多标签算法的数据测试，有时当样本极少时（比如yeast只有8个样本）比样本偏大更加花费建模时间，这是因为标签太多（比如yeast一共有103个样本）。折腾导入Mulan、weka包，以及去更改源码中的算法参数花了些时间。总体来说多标签算法比较难。