**基于WEKA的数据挖掘算法学习（七）**

# 特征选择

## 1.1子集搜索与评价

给定属性集，我们将属性称为“特征”，“相关属性”：对当前学习任务有用的属性，“无关属性”：对当前学习任务没什么用的属性。从给定的特征集合中选择相关特征子集的过程，称为“特征选择”。

特征选择是一个重要的数据预处理过程，在现实机器学习任务中，获得数据之后通常进行特征选择，然后再训练学习器。原因：1、解决维数灾难问题；2、留下关键因素，数据真相更加明确。但需要保证特征选择过程必须确保不丢失重要特征，否则后续学习过程会因为重要信息缺失而无法获得好的性能。

两个关键的环节：

1、“子集搜索”：给定特征集合，将每个特征看作一个候选子集，对候选单特征子集进行评价。直到第k+1轮时，最优的候选(k+1)特征子集不如上一轮的选定集。这称为“前向搜索”，还可以使用“后向搜索”和“双向搜索”。

2、“子集评价”：对属性子集A，假定根据其取值将数据集D分成了V个子集，每个子集中的样本在A上取值相同，于是计算到属性子集A的信息增益：

其中信息熵定义为：。信息增益越大，意味着特征子集A包含的有助于分类信息越多。

将特征子集搜索机制与子集评价机制相结合即可得到特征选择方法，大致分为三类：过滤式(filter)、包裹式(wrapper)、嵌入式(embedding)。

## 1.2 过滤式选择

先对数据集进行特征选择，然后训练学习器，特征选择过程与后续学习器无关。先用特征选择过程对初始特征进行“过滤”，再用过滤后的特征来训练模型。

Relif方法是著名的过滤式特征选择方法，设计了“相关统计量”来度量特征的重要性。该统计量是一个向量，其每个分量分别对应一个初始特征，而特征的重要性由子集中每个特征所对应的相关统计量分量之和来决定。可以指定阈值，也可以指定预选取的的特征个数K。Relief的关键是确定相关统计量，通常使用猜中近邻和猜错近邻的结合。对基于不同样本得到的估计结果进行平均，就得到各属性的相关统计量的分量，分量值越大，则对应属性的分类能力就越强。时间开销随采样次数以及原始特征数线性增长，因此是一种运行效率很高的过滤式特征选择算法。

## 1.3 包裹式选择

过滤式特征选择不考虑后续学习器的不同，包裹式特征选择直接把最终要使用的学习器的性能作为特征子集的评价标准。即为给定学习器选择最有利于其性能、“量身定做”的特征子集。

包裹式特征选择方法直接针对给定学习器进行优化，从最终性能来说，比过滤式特征选择更好，但在特征选择过程中需多次训练学习器，因此包裹式特征选择的计算开销通常比过滤式特征选择大很多。

LVW是典型的包裹式特征选择方法，在拉斯维加斯方法框架下使用随机策略进行子集搜索，以最终分类器的误差为特征子集评价准则。由于特征子集搜索采用了随机策略，而每次特征子集评价都需训练学习器，计算开销很大，因此算法需要设置停止条件控制参数T。

## 1.4嵌入式选择

在过滤式和包裹式特征选择方法中，特征选择过程与学习器训练过程有明显的分别。而嵌入式特征选择过程与学习器训练过程融为一体，即学习器训练过程中自动地进行了特征选择。

给定数据集D，考虑最简单的线性回归模型，以平方误差为损失函数，则优化目标为：

当将范数替换为，令P=1，即采用范数，则有LASSO：

二者均可降低过拟合风险，但是范数更易获得稀疏解，即求得的w会有更少的非零分量。

# 2.weka算法实现

## 2.1数据集的准备

1、UCI完备数据集：pima-indians-diabetes.data.csv.arff

2、UCI不完备数据集：yeast.data.csv.arff

3、UCI不平衡数据集： car.csv.arff

## 2.2 手工特征选择

选择Classifyr🡪J48

设置confidenceFactor = 0.25，设置minNumObj=2，这里选择UCI完备数据集

pima-indians-diabetes来进行手工属性选择数据测试，结果如下所示。

**（1）**首先，pima-indians-diabetes一共有9个属性，如下展示：

1. Number of times pregnant

2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test

3. Diastolic blood pressure (mm Hg)

4. Triceps skin fold thickness (mm)

5. 2-Hour serum insulin (mu U/ml)

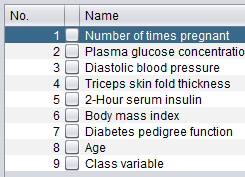
6. Body mass index (weight in kg/(height in m)^2)

7. Diabetes pedigree function

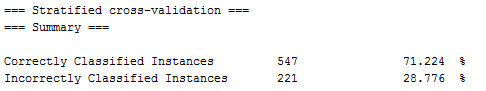
8. Age (years)

9. Class variable (0 or 1)

**（2）**添加所有属性进行J48测试：

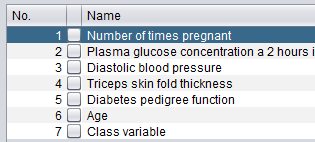


**图1.1添加所有属性**

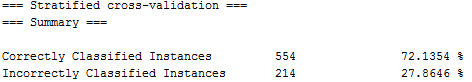


**图1.2分类正确率71.224 %**

**（3）**remove两个属性后进行J48测试：

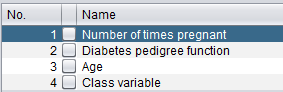


**图2.1添加7个属性**

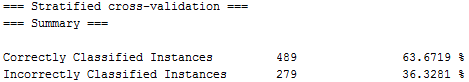


**图2.2分类正确率72.1354 %**

**（4）**remove五个属性后进行J48测试：



**图3.1添加4个属性**



**图3.2分类正确率63.6719 %**

（5）总结：

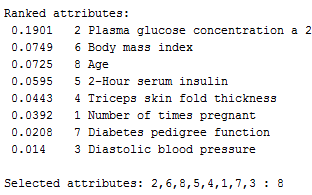
由上述手动属性选择可以看见，同一数据集pima-indians-diabetes在属性数目添加不同的情况下，得到不同的结果，且当删除某些属性后，分类正确率有提高，更多的属性反而会降低准确率。也印证了前面介绍的无关属性对当前学习任务没有什么用。

此外，这里只展示了部分属性的remove，在属性为9的情况下，若全部进行实验测试，一共需要完成的实验次数为：（次），512次实验不可能全部手动跑完，因此这里不做完整展示。

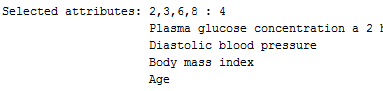
## 2.3 自动特征选择、由J48测试特征选择效果

### 2.3.1 pima

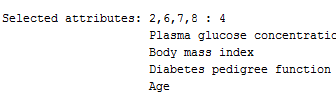
1、选择Select attributes🡪**InfoGainAttributeEval、Ranker**🡪Use full trainging set，特征选择结果为：选择属性为2,6,8,5,4,1,7,3（按重要程度排列，测试时选择前五个特征，后三个特征重要程度偏低）



2、选择Select attributes🡪 **WrapperSubsetEva（学习器为J48）、BestFirst** 🡪Use full trainging set，特征选择结果为：选择属性为2,3,6,8



3、选择Select attributes🡪 **CfsSubsetEval、GreedyStepwise** 🡪Use full trainging set，特征选择结果为：选择属性为2,6,7,8



4、根据三种特征选择算法的不同属性选择，使用选择选择Classifyr🡪J48设置confidenceFactor = 0.25，设置minNumObj=2，选择五折交叉验证，进行分类结果对比：

|  |  |
| --- | --- |
| 特征选择算法 | J48分类正确率 |
| 未进行属性选择 | 71.224% |
| InfoGainAttributeEval(属性选择：2,6,8,5,4) | 73.307% |
| **WrapperSubsetEva(属性选择：2,3,6,8)** | **76.042%** |
| CfsSubsetEval（属性选择：2,6,7,8） | 72.40% |

由上表结果可得，完备数据集pima在使用包装方法选择WrapperSubsetEva评估器，特征选择为2,3,6,8时，得到效果最佳，分类正确率最高，属性子集虽然大大缩小，但效果却更好。且三种测试的特征选择算法结果均比未进行属性选择时更加优越，也看出特征选择的重要性。

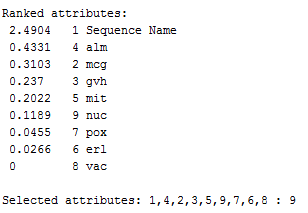
5、根据三种特征选择算法的不同属性选择，使用选择选择Cluster🡪SimpleKMeans设置numCluster=2，进行聚类结果对比：

|  |  |
| --- | --- |
| 特征选择算法 | KMeans聚类errors |
| 未进行属性选择 | 149.5177 |
| InfoGainAttributeEval(属性选择：2,6,8,5,4) | 86.7654 |
| **WrapperSubsetEva(属性选择：2,3,6,8)** | **72.1875** |
| CfsSubsetEval（属性选择：2,6,7,8） | 67.8501 |

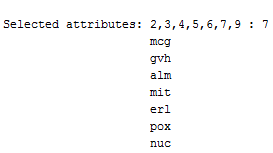
在上周的聚类算法报告里面有写过，KMeans算法中Within cluster sum of squared errors越小，表明聚类效果越好。这里看到，进行属性选择后，Within cluster sum of squared errors都有减小，其中，CfsSubsetEval算法让errors达到了67.8501，性能有明显提升。

### 2.3.2 yeast

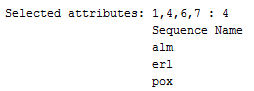
1、选择Select attributes🡪**InfoGainAttributeEval、Ranker**🡪Use full trainging set，特征选择结果为：选择属性为1,4,2,3,5,9,7,6,8（按重要程度排列，测试时选择前六个特征，后三个特征重要程度偏低）



2、选择Select attributes🡪 **WrapperSubsetEva（学习器为J48）、BestFirst** 🡪Use full trainging set，特征选择结果为：选择属性为2,3,4,5,6,9



3、选择Select attributes🡪 **CfsSubsetEval、GreedyStepwise** 🡪Use full trainging set，特征选择结果为：选择属性为1,4,6,7



4、根据不同属性选择，使用选择选择Classifyr🡪J48设置confidenceFactor = 0.25，设置minNumObj=2，选择五折交叉验证，进行分类结果对比：

|  |  |
| --- | --- |
| 属性选择分类 | J48分类正确率 |
| 未进行属性选择 | 50.404% |
| InfoGainAttributeEval(属性选择：1,4,2,3,5,9) | 49.798% |
| **WrapperSubsetEva(属性选择：2,3,4,5,6,9)** | **56.132%** |
| CfsSubsetEval（属性选择：1,4,6,7） | 41.03% |

由上表结果可得，不完备数据集yeast在使用包装方法选择WrapperSubsetEva评估器，特征选择为2,3,4,5,6,9时，得到效果最佳，分类正确率最高。但InfoGainAttributeEval和CfsSubsetEval特征选择方法并未使性能有所提高，反而下降。

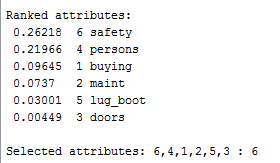
5、根据三种特征选择算法的不同属性选择，使用选择选择Cluster🡪SimpleKMeans设置numCluster=2，进行聚类结果对比：

|  |  |
| --- | --- |
| 特征选择算法 | KMeans聚类errors |
| 未进行属性选择 | 2428.8527 |
| InfoGainAttributeEval(属性选择：1,4,2,3,5,9) | 2188.4318 |
| **WrapperSubsetEva(属性选择：2,3,4,5,6,9)** | **722.2902** |
| CfsSubsetEval（属性选择：1,4,6,7） | 2112.5947 |

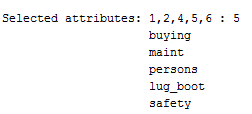
。这里看到，进行属性选择后，Within cluster sum of squared errors都有减小，其中，WrapperSubsetEva算法让errors达到了722.2902，相比于未进行属性选择性能大大提升，另外两种特征选择算法虽然降低errors值不明显，但也有降低，且时间效率上来说，属性减少，建模耗费时间也相应降低。

### 2.3.3 car

1、选择Select attributes🡪**InfoGainAttributeEval、Ranker**🡪Use full trainging set，特征选择结果为：选择属性为6,4,1,2,5,3（按重要程度排列，测试时选择前四个特征，后二个特征重要程度偏低）



2、选择Select attributes🡪 **WrapperSubsetEva（学习器为J48）、BestFirst** 🡪Use full trainging set，特征选择结果为：选择属性为1,2,4,5,6



3、选择Select attributes🡪 **CfsSubsetEval、GreedyStepwise** 🡪Use full trainging set，特征选择结果为：选择属性为6



4、根据不同属性选择，使用选择选择Classifyr🡪J48设置confidenceFactor = 0.25，设置minNumObj=2，选择五折交叉验证，进行分类结果对比：

|  |  |
| --- | --- |
| 属性选择分类 | J48分类正确率 |
| 未进行属性选择 | 91.551% |
| InfoGainAttributeEval(属性选择：6,4,1,2) | 86.111% |
| WrapperSubsetEva(属性选择：1,2,4,5,6) | **91.956%** |
| CfsSubsetEval（属性选择：6） | 70.023% |

由上表结果可得，不平衡数据集car在使用包装方法选择WrapperSubsetEva评估器，特征选择为1,2,4,5,6时，得到效果最佳，分类正确率最高。

5、根据三种特征选择算法的不同属性选择，使用选择选择Cluster🡪SimpleKMeans设置numCluster=2，进行聚类结果对比：

|  |  |
| --- | --- |
| 特征选择算法 | KMeans聚类errors |
| 未进行属性选择 | 6577.0 |
| InfoGainAttributeEval(属性选择：6,4,1,2) | 4238.0 |
| WrapperSubsetEva(属性选择：1,2,4,5,6) | 5282.0 |
| **CfsSubsetEval（属性选择：6）** | **449.0** |

根据这里结果展示可以知道，未进行属性选择时，用KMeans聚类时errors值达到了6577.0，使用后均有提升，特别是CfsSubsetEval的效果尤为显著。

### 2.3.4 总结

根据三种特征选择算法测试完备、不完备、不平衡数据集结果来看，总有特征选择算法在删除掉一些无关属性后，在使用J48测试效果时，使得分类结果更佳。

其中，由于包裹式特征选择使用的学习器为J48，也就是特征选择为J48量身定做，所以当我用J48进行测试特征选择的优劣时，三种数据集均在WrapperSubsetEva算法这里算法结果有提升，因此，个人觉得包裹式选择很实用，只是时间效率上来说不太好。

在使用聚类算法KMeans来测试时，进行特征选择后，性能均有明显提升，只是有的数据集在使用某些特征选择算法时，效果更好。但是整体性能都有所提升，且随着属性的减少，可以明显感受到建模时间大大缩短。

因此，印证前面理论知识介绍所说：特征选择是一个重要的数据预处理过程，在现实机器学习任务中，获得数据之后通常进行特征选择，然后再训练学习器，带来好处：1、解决维数灾难问题；2、留下关键因素，数据真相更加明确。特别是当属性数目特别大的时候，这种数据预处理的好处更明显。

# 3.本周学习心得体会

本周学习了《机器学习 周志华版》中的特征选择与稀疏学习，对过滤式选择、包裹式选择、嵌入式选择理论基础有了一定了解。过滤式选择用“过滤”后的特征来训练模型；包裹式选择为给定学习器选择最优性能的特征子集；嵌入式选择与训练过程融为一体。个人更偏向于包裹式选择，虽然时间效率上欠佳，但是能很好的适应所选择的学习器。总的来说不算太难。