西北工業大學

NORTHWESTERN POLYTECHNICAL UNIVERSITY

信息检索与数据挖掘

实验指导

计算机学院

2017 年 03 月

修订记录

版本	修订日期	修订者	修订内容
1.0	2008/12	张利军	
2.0	2009/11	张利军	添加附录 Weka 入门教程
3.0	2014/10	张利军	1. 根据新的实验安排，将实验内容分为 5 部分。 2. Weka 版本由 3.6.1 变更为 3.6.11
4.0	2016/3	张利军	1. 添加信息检索相关实验 2. 章节目录调整
4.1	2017/3	张利军	取消可选实验，全部设置为必选。

目录

1. 实验概要

1.1. 实验说明

1.1.1. 内容

本课程实验分两部分，数据挖掘和信息检索，各 5 个实验，每个人必须完成全部实验。实验课共 16 学时，在课堂内不能完成的，需利用课余时间完成。具体实验内容参考本指导的其余部分。

1.1.2. 报告

每个实验需要撰写实验报告，实验报告的内容包括：实验内容、实验步骤、运行结果（可以是程序的输出，也可以是运行画面的抓屏，抓屏图片要尽可能的小，否则文件太大）。每份实验报告是一个 word 文档。实验报告命名规则如下：

IRDMX.X_YYYYYYYYYYZZZ.doc

其中 X.X 表示该实验的编号，对应于本指导中实验的章节编号，如第二个实验：分类的编号是 2.2，YYYYYYYYYY 是学号，ZZZ 是姓名。如：学号为 2012302675 的学生的分类实验的报告文件名为：

IRDM2.2_2012302675 张霸天.doc

实验完成后，将电子版的实验报告发送到 npu_irdm@126.com 邮箱，如果该次实验需要编写程序，需要把源程序文件(.java,.c 或其它文件)、数据文件(如果有的话)和可执行文件(如果有的话)一起发送到以上邮箱，由实验辅导老师批改打分。期末统一提交到学院归档。

注意：请每个人保存好自己的实验报告以及源代码，直到该门课成绩公布之后。

1.1.3. 考核

实验总成绩为100分，考勤和课堂表现30%，实验报告70%。每次实验的报告按100分计算。操作行验证类实验主要考察每次实验的内容是否都完成，每一步是否正确，是否得到合理的实验结果。编程设计型题目主要考察程序功能是否实现，运行是否正确，是否可得到期望的结果，代码是否规范，报告文档是否规范等。

1.2. 实验环境和配置

OS：Windows 2000/XP/Win7

DM Tool：Weka3.6.11

IR：Lucene

IDE：Eclipse

1.3. 实验课要求

1．实验课上只允许做本课程实验相关的工作，请勿做其它事情，更不能游戏、网络聊天、浏览不相关网站等，请遵守机房的相关规定。

2．实验之前，最好提前预习所要做的实验，大致了解本次实验需要做的内容。

3．实验课时比较少，实验内容相对较多，课内无法完成的需大家在课后完成。

2. 数据挖掘

2.1. 数据预处理和关联分析

2.1.1. 目的和要求

1. 掌握如何下载及安装 Weka。
2. 了解 Weka 的数据文件格式。
3. 掌握在 Weka 的 Explorer 中对数据预处理。

2.1.2. 实验准备

参考资料：本文档附录三的内容。

2.1.3. 实验内容

1. 下载并安装 Weka （10 分）
   从以下站点下载新版 Weka 并安装
   http://www.cs.waikato.ac.nz/ml/weka/downloading.html

2. 将以下的数据转换为 ARFF 格式（20 分）
   1) 手工进行转换
   2) 利用 Weka 的命令行进行转换
   3) 利用 Explorer 进行转换

   提示：先将其保存为 csv 文件，然后转换为 ARFF 文件

id	*age*	*income*	*student*	*credit_rating*	*buys_computer*
1001	30	high	no	fair	no
1002	25	high	no	excellent	no
1003	32	high	no	fair	yes
1004	55	medium	no	fair	yes
1005	41	low	yes	fair	yes

1006	46	low	yes	excellent	no
1007	32	low	yes	excellent	yes
1008	24	medium	no	fair	no
1009	21	low	yes	fair	yes
1010	53	medium	yes	fair	yes
1011	28	medium	yes	excellent	yes
1012	33	medium	no	excellent	yes
1013	36	high	yes	fair	yes
1014	50	medium	no	excellent	no

3. 在 Explorer 中对以上数据进行预处理。（35 分）
   1) 将无用属性 id 删除。
   2) 将 age 属性离散化为 youth，midage，oldage 三个区间。

4. 关联规则挖掘.（35 分）
   1) 在 Explorer 中用 Apriori 算法对上述数据生成置信度最高的 15 条关联规则，设最小支持度为 0.2，最小置信度为 0.7
   2) 使用 Weka 的命令行执行以上任务。

2.2. 分类

2.2.1. 目的和要求

1. 掌握在 Weka 的 Explorer 中对数据进行分类。

2.2.2. 实验准备

参考资料：本文档附录三的内容。

2.2.3. 实验内容

分类

1) 利用 C4.5 决策树和 10 折交叉验证法对附件 bank-data.csv 中的数据进行分类并评估其准确率。(30 分)

2) 观察 1）中所得到的树模型,如果模型过于复杂或者过于简单请调整剪枝参数以得到一个合适的树模型。(20 分)

3) 利用 NaiveBayes 和 HoldOut 方法对附件 bank-data.csv 中的数据进行分类并评估其准确率。(50 分)

*提示：使用 Holdout 方法时需要首先将其分为训练集和测试集两部分，最好训练集 VS 测试集为 2：1，注意划分时要**随机**划分。*

2.3. 回归和聚类分析

2.3.1. 目的和要求

1. 掌握在 Weka 的 Exploer 中对数据进行回归分析。
2. 掌握在 Weka 的 Exploer 中对数据进行聚类分析。

2.3.2. 实验准备

参考资料：本文档附录三的内容。

2.3.3. 实验内容

1. 在 Explorer 中进行回归分析。（40 分）

   利用附件 Concrete_Data.xls 中的数据预测下表中混凝土的抗压强度。

   提示：可以先对附件中的数据进行线性回归分析，建立模型，然后用模型进行预测。

Cement	***Blast Furnace Slag***	***Fly Ash***	***Water***	***Superplasticizer***	***Coarse Aggregate***	***Fine Aggregate***	***Age***	***Concrete compressive strength***
260	95	0	228	0	932	594	93	
427.5	87.5	0	228	10.1	801	594	7	
385	0	17	146	0	1120	800	38	
531.3	0	0	141.8	28.2	852.1	893.7	28	
418.8	212.5	0	155.7	14.3	852.1	880.4	53	
405.8	84.7	12.5	147.4	11.4	946.8	852.1	91	
333.8	0	94.6	197.9	4.6	947	852.2	30	
112.1	0	81.6	180.3	5.7	1057.6	779.3	48	
113.8	98.1	24.5	181.7	6.7	1066	701.5	102	
170.3	32	125.2	161.9	9.9	1088.1	802.6	18	

2. 在 Explorer 中进行聚类分析。（60 分）

利用 k-means 方法对附件 water-treatment.data 中的数据进行聚类分析。设要将数据分类到 10 个簇中。调整参数 seed 的值，以使得聚类的平方误差和较小。将聚类结果保存为文件 water-treatment_clustered.arff。

2.4. 挖掘综合运用

2.4.1. 目的和要求

掌握利用数据挖掘工具 Weka 解决现实问题的综合方法。

2.4.2. 实验准备

参考资料：本文档附录三的内容。

2.4.3. 实验内容

Movielens 中的数据是关于观众对电影的评分情况，请根据附件中的数据，在 Weka 中建立一个模型，使得这个模型能够预测观众对某部影片的打分。

1) 数据预处理：（40 分）

a) 将原始数据中的缺失值用‘？’替换，原始数据中的逗号用‘$’替换(以免生成 csv 文件后产生错误).

提示：在使用‘|’作为分隔符的文件中所有形为“||”的值为缺失值，应该被替换为“|?|”.

b) 利用附件中的 mergemovielens 工具将 u.user,u.item,u.data 合并为 movielens.csv 文件.

注意：在合并后的文件中，邮政编码只取了第一位（第一位表示州）。

c) 在 Weka 中打开 movielens.csv 文件进行预处理.

提示:如果数据量太大,产生堆栈溢出错误,可以修改 Weka 安装目录下 RunWeka.ini 文件中 maxheap 的值，使得 Weka 能够将 movielens.csv 装载进内存中。

d) 删除无用属性 userid 和 movieid。为了处理方便，我们认为 movietitle，releasedate, url 和 timestamp 对预测没有意义，将这 4 个属性也删除。

e) 将 age 属性离散化为三个区间，分别用 youth，middle_aged 和 senior 来替换。将得到的文件保存为 movielens.arff.

f) 将 movielens.arff 中所有电影流派属性（包括 unknown）改为取值为{0，1}的标称型变量。

g) 利用 InfoGainAttributeEval 作为属性选择算法，选取信息增益大于 0.0005 的属性，其余属性删除。

提示：在使用这种属性选择算法前，需要将 rating 分类属性的类型改为分类型，否则不能够使用这种算法。

h) 保存该数据文件为 movielens.arff，并且将关系名称改为“movielens”。

2) 模型建立：选择合适的预测模型(分类或回归)，必要的参数的选择等。(40 分)

a) 使用 NaiveBayes 方法建立分类器模型；

b) 使用决策树方法建立分类器模型；

c) 使用线性回归方法建立预测模型；

3) 模型评估：如何分割数据集进行评估，以及选择什么样的评估标准对预测模型进行评估：准确率，误差等。(20 分)

对 2)中建立的模型记录它们的准确率，或者误差，看看哪个模型性能更好一些。思考一下用什么方法能提高其准确率或者降低误差。

2.5. 挖掘算法编程

2.5.1. 目的与要求

设计并实现一个挖掘算法。

2.5.2. 实验准备

略

2.5.3. 实验内容

附件中 contact-lenses 中的数据是关于眼光师针对病人的情况作出的诊断：使用软的隐形眼镜，硬的隐形眼镜或不能佩戴隐形眼镜。根据以下要求，编写程序，针对某未知病人作出应该佩戴什么类型隐形眼睛的诊断。编程语言不限。请提交源程序和最终的可执行程序。

1. 朴素贝叶斯。利用朴素贝叶斯方法根据 train.csv 文件中的数据进行学习，建立贝叶斯分类模型。然后利用该模型对未知数据进行分类。

```
bayeslearner train.csv bayes.model
bayesclassfier bayes.model test.csv result.csv
```

其中，train.csv 和 test.csv 分别为训练数据和测试数据，其格式参见附件中的数据。bayes.model 用来保存建立的模型，格式自己设计。result.csv 用来保存对 test.csv 中的未知数据分类后的结果，格式自己设计。Bayeslearner 是一个学习器，用来从 train.csv 学习(训练)分类器模型，其输出为 bayes.model。bayesclassfier 是一个分类器，用来对 test.csv 中的数据进行分类，其输出为 result.csv。

2. K-最近邻居分类法，用 k-最近邻居法对 train.csv 中的数据进行分类。

```
knnclassifier k train.csv test.csv result.csv
```

其中 k 为 k-nn 中的参数，表示对每个未知元组要计算它的 k 个最相近的邻居。train.csv test.csv result.csv 与 1 中的含义相同。

3. 信息检索

3.1. 倒排文档索引

3.1.1. 目的和要求

1．了解倒排文档的结构和建立倒排文档索引的方法；

2．掌握 Lucene 索引中 API 的使用。

3.1.2. 实验准备

1．了解倒排文档索引相关知识。

2．了解 Lucene，并掌握 Lucene 中索引相关 API 的用法。

3．掌握 Eclipse 的安装，配置和用法。

3.1.3. 实验内容

1．安装所需的软件和类库，并配置开发环境；

2．在 D:\IR\Collections 文件夹中放入 text,word,pdf 文件；

3．在 Eclipse 中创建工程，编写程序，调用 Lucene 中的索引创建 API，对 D:\IR\Collections 中的文件创建索引，索引文件保存在 D:\IR\Index 文件夹中。

4．观察 D:\IR\Index 中是否已经创建好索引。

3.2. 词干提取和停用词过滤

3.2.1. 目的和要求

1．了解词干提取和停用词过滤在搜索引擎中的作用，分析搜索引擎使用词干提取和停用词过滤的情况；

2．掌握 Lucene 中词干提取和停用词过滤分析器的使用。

3.2.2. 实验准备

1．了解词干提取和和停用词过滤的概念；

2．了解 Lucene，学习 Lucene 中词干提取和停用词过滤相关 API 的用法。

3.2.3. 实验内容

1．分别在 Baidu 和 Google(无法使用 Google 可以使用任何其它的搜索引擎，如 360 搜索等)中进行测试，构建有代表性的查询，验证各个搜索引擎如何处理词干提取和停用词过滤；

1) 不同的搜索引擎中查询“瓷器和玉器”，查看搜索结果，验证对应的搜索引擎是否过滤了停用词（本例中是“和”）。
2) 搜索引擎中查询“瓷器和玉器”（查询带引号），查看搜索结果，验证和上一步的结果有何不同。
3) 不同的搜索引擎中查询“data mine”，查看搜索结果，验证对应的搜索引擎是否进行了词干提取（可以搜索到不同的 mine 形式）

2．调用 Lucene API 编写程序，在构建索引和检索时进行词干提取和停用词过滤。

1) 编写程序，调用 LuceneAPI 对以下的文本进行构建索引，构建索引时需要进行词干提取和索引过滤。
2) 使用文本编辑器打开索引文件夹下 tis 后缀的文件，查看其中的此项是否经过词干提取。如下文中有单词“singing”,“drunkenly ”“soldiers”，在 tis 文件中对应的词项是什么。
3) 编写程序，调用 LuceneAPI 从上步创建的索引中进行检索，分别对以下查询进行查询，查看是否能得到正确的结果：

   查询 1：“singing”

   查询 2：“drunkenly”

查询 3：“soldiers”

查询 4：“soldiers has”

查询 5：“singing and”

Casablanca

There is a scene about halfway through the movie Casablanca that has become commonly known as 'The Battle of the Anthems' throughout the film's long history. A group of German **soldiers** has come into Rick's Café American and are **drunkenly** singing the German National Anthem at the top of their voice. Victor Lazlo, the leader of the French Resistance, cannot stand this act and while the rest of the club stares appalled at the Germans, Lazlo orders the band to play 'Le Marseilles (sic?)' the French National Anthem. With a nod from Rick, the band begins playing, with Victor singing at the top of HIS voice. This in turn, inspires the whole club to begin **singing** and the Germans are forced to surrender and sit down at their table, humbled by the crowd's dedication. This scene is a turning point in the movie, for reasons that I leave to you to discover.

3.3. 中文分词

3.3.1. 目的和要求

1．了解信息检索中分词的原理、中英文分词的不同、中文分词的特点以及分词对检索的作用；

2．掌握如何编写程序，在 Java 中调用不同的分词工具包对中文文本进行分词。

3.3.2. 实验准备

1．调查有哪些中文分词工具可以利用；

2．深入了解 NLPIR（原名 ICTCLAS）中文分词组件的用法。

3．深入了解 IKAnalyzer 中文分词工具包的用法。

3.3.3. 实验内容

1．从 http://ictclas.nlpir.org/站点下载 NLPIR 中文分词组件。

2．在 Eclipse 中创建工程，编写程序，调用 NLPIR，对以下文本进行分词。

> 微软是一家总部位于美国的跨国科技公司
> 共同创造美好的新世纪
> 工信处女干事每月经过下属科室都要交代交换机的维护工作
> 南京市长江大桥

3．输出分词结果，观察分词结果如何。

4．从 http://git.oschina.net/wltea/IK-Analyzer-2012FF 下载 IK Analyzer 分词组件包。

5．在 Eclipse 中编写程序，调用 IK Analyzer 对上述的文本进行分词。

6．输出分词结果，观察结果如何，并与 NLPIR 的分词结果对比，看看有何不同。

3.4. 向量空间模型

3.4.1. 目的和要求

1．通过实验深化理解信息检索中向量空间模型的主要思想；
2．了解常见的向量空间模型的计算公式和方法；
3．利用程序实现向量空间模型并将其应用于信息检索。

3.4.2. 实验准备

1．查阅相关资料了解向量空间模型；
2．查阅相关资料掌握词项权重的计算方法，查询和文档相关性计算方法。

3.4.3. 实验内容

1．在 Eclipse 中编写一个类，可以计算文档内部索引词的词频；
2．编写一个类，可以计算索引词的权重；
3．编写一个类，可以计算一个查询和文档的相似度；
4．使用编写的向量空间计算程序对检索结果进行排序；
5．利用 Lucene 默认的向量空间模型进行检索。

3.5. PageRank

3.5.1. 目的和要求

1．深化理解 PageRank 算法的主要思想；
2．熟悉 PageRank 的计算公式和方法；

3.5.2. 实验准备

查阅相关资料，了解 PageRank 算法的主要思想和计算公式。

3.5.3. 实验内容

1．使用 Java 编写一个计算 PageRank 算法的程序

4. 附录一 实验报告格式

《信息检索与数据挖掘》实验报告				
实验题目： **数据预处理与关联分析**	**姓名**	**班级**	**学号**	**日期**

实验内容和步骤结果

1. 下载并安装 Weka
   运行界面：

2. 将以下的数据转换为 ARFF 格式。
   具体步骤：
   转换结果：

3. 在 Explorer 中对以上数据进行预处理。
   处理过程：
   处理结果：

4. 关联规则挖掘。
   参数设定：
   挖掘结果：

5. 实验中出现的问题以及解决方案(对于未解决问题请将问题列出来)
   除了标题内容以外，该部分内容中还可以写对于实验的一些感受，建议，意见等。

批阅者：
批阅日期：
实验成绩：
批注：

5. 附录二 实验数据的说明

5.1. concrete

concrete 中的数据表示的是混凝土中各种成分和混凝土抗压强度之间的关系。总共 9 个属性，全部为连续型值。分别为：

Cement (component 1)(kg in a m^3 mixture)：水泥含量，每立方混凝土中的含量。单位为 kg。

Blast Furnace Slag (component 2)(kg in a m^3 mixture)：高炉熔碴，每立方混凝土中的含量，单位为 kg。

Fly Ash (component 3)(kg in a m^3 mixture)：飞尘，每立方混凝土中的含量，单位为 kg。

Water (component 4)(kg in a m^3 mixture)：水，每立方混凝土中的含量，单位为 kg。

Superplasticizer (component 5)(kg in a m^3 mixture)：超增塑剂，每立方混凝土中的含量，单位为 kg。

Coarse Aggregate (component 6)(kg in a m^3 mixture)：粗沙，每立方混凝土中的含量，单位为 kg。

Fine Aggregate (component 7)(kg in a m^3 mixture)：细集料混合物，每立方混凝土中的含量，单位为 kg。

Age (day)：凝固时间，单位为天。

Concrete compressive strength(MPa, megapascals)：抗压强度，该属性为要预测的属性（分类属性），单位为 Mpa。

5.2. contact-lenses

contact-lenses 中的数据是关于眼光师针对病人的情况作出的诊断。共 5 个属性，全部为分类型属性。分别为：

age：年龄，三种取值，young 年轻，pre-presbyopic，presbyopic 老花眼。

spectacle-prescrip：视力诊断，myope 近视，hypermetrope 远视。

astigmatism：散光，no，yes。

tear-prod-rate：泪流量，reduced 较少，normal 正常。

contact-lenses：推荐镜片，该属性为分类属性。soft 软镜片，hard 硬镜片，none 不推荐佩戴隐形眼镜。

5.3. movielens

6. 附录三 Weka 入门教程[1]

6.1. 简介

WEKA 的全名是怀卡托智能分析环境（Waikato Environment for Knowledge Analysis），它的源代码可通过 http://www.cs.waikato.ac.nz/ml/weka 得到。同时 weka 也是新西兰的一种鸟名，而 WEKA 的主要开发者来自新西兰。

WEKA 作为一个公开的数据挖掘工作平台，集合了大量能承担数据挖掘任务的机器学习算法，包括对数据进行预处理，分类，回归、聚类、关联规则以及在新的交互式界面上的可视化。

如果想自己实现数据挖掘算法的话，可以看一看 weka 的接口文档。在 weka 中集成自己的算法甚至借鉴它的方法自己实现可视化工具并不是件很困难的事情。

2005 年 8 月，在第 11 届 ACM SIGKDD 国际会议上，怀卡托大学的 Weka 小组荣获了数据挖掘和知识探索领域的最高服务奖，Weka 系统得到了广 泛的认可，被誉为数据挖掘和机器学习历史上的里程碑，是现今最完备的数据挖掘工具之一（已有 11 年的发展历史）。Weka 的每月下载次数已超过万次。

6.2. 数据格式

巧妇难为无米之炊。首先我们来看看 WEKA 所用的数据应是什么样的格式。

跟很多电子表格或数据分析软件一样，WEKA 所处理的数据集是图 1 那样的一个二维的表格。

[1] 本章内容来自网络

weather.arff

Relation: weather

No.	1.outlook Nominal	2.temperature Numeric	3.humidity Numeric	4.windy Nominal	5.play Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	72.0	90.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	71.0	91.0	TRUE	no

图 1

这里我们要介绍一下 WEKA 中的术语。表格里的一个横行称作一个实例（Instance），相当于统计学中的一个样本，或者数据库中的一条记录。竖行称作 一个属性（Attribute），相当于统计学中的一个变量，或者数据库中的一个字段。这样一个表格，或者叫数据集，在 WEKA 看来，呈现了属性之间的一种 关系(Relation)。图 1 中一共有 14 个实例，5 个属性，关系名称为“weather”。

WEKA 存储数据的格式是 ARFF（Attribute-Relation File Format）文件，这是一种 ASCII 文本文件。图 1 所示的二维 表格存储在如下的 ARFF 文件中。这也就是 WEKA 自带的“weather.arff”文件，在 WEKA 安装目录的“data”子目录下可以找到。

代码:

```
% ARFF file for the weather data with some numric features
%
@relation weather    http://www.chinakdd.com

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
```

%
% 14 instances
%
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no

需要注意的是，在 Windows 记事本打开这个文件时，可能会因为回车符定义不一致而导致分行不正常。推荐使用 UltraEdit 这样的字符编辑软件察看 ARFF 文件的内容。

下面我们来对这个文件的内容进行说明。

识别 ARFF 文件的重要依据是分行，因此不能在这种文件里随意的断行。空行（或全是空格的行）将被忽略。

以“%”开始的行是注释，WEKA 将忽略这些行。如果你看到的“weather.arff”文件多了或少了些“%”开始的行，是没有影响的。

除去注释后，整个 ARFF 文件可以分为两个部分。第一部分给出了头信息（Head information），包括了对关系的声明和对属性的声明。第二部 分给出了数据信息（Data information），即数据集中给出的数据。从“@data”标记开始，后面的就是数据信息了。

关系声明

关系名称在 ARFF 文件的第一个有效行来定义，格式为

@relation <relation-name>

<relation-name>是一个字符串。如果这个字符串包含空格，它必须加上引号（指英文标点的单引号或双引号）。

属性声明

属性声明用一列以“@attribute”开头的语句表示。数据集中的每一个属性都有它对应的“@attribute”语句，来定义它的属性名称和数据类型。

这些声明语句的顺序很重要。首先它表明了该项属性在数据部分的位置。例如，“humidity”是第三个被声明的属性，这说明数据部分那些被逗号分开的列 中，第三列数据 85 90 86 96 ... 是相应的“humidity”值。其次，最后一个声明的属性被称作 class 属性，在分类或回归任务 中，它是默认的目标变量。

属性声明的格式为

@attribute <attribute-name> <datatype>

其中<attribute-name>是必须以字母开头的字符串。和关系名称一样，如果这个字符串包含空格，它必须加上引号。

WEKA 支持的<datatype>有四种，分别是

numeric	数值型
<nominal-specification>	分类（nominal）型
string	字符串型
date [<date-format>]	日期和时间型

其中<nominal-specification> 和<date-format> 将在下面说明。还可以使用两个类型“integer”和“real”，但是 WEKA 把它们都当作“numeric”看待。注意 “integer”，“real”，“numeric”，“date”，“string”这些关键字是区分大小写的，而 “relation”“attribute ”和“date”则不区分。

数值属性

数值型属性可以是整数或者实数，但 WEKA 把它们都当作实数看待。

分类属性

分类属性由<nominal-specification>列出一系列可能的类别名称并放在花括号中：{<nominal- name1>, <nominal-name2>, <nominal-name3>, ...} 。数据集中该属性的值只能是其中一种类别。

例如如下的属性声明说明“outlook”属性有三种类别：“sunny”，“ overcast”和“rainy”。而数据集中每个实例对应的“outlook”值必是这三者之一。

@attribute outlook {sunny, overcast, rainy}

如果类别名称带有空格，仍需要将之放入引号中。

字符串属性

字符串属性中可以包含任意的文本。这种类型的属性在文本挖掘中非常有用。

示例：

@ATTRIBUTE LCC string

日期和时间属性

日期和时间属性统一用“date”类型表示，它的格式是

@attribute <name> date [<date-format>]

其中<name>是这个属性的名称，<date-format>是一个字符串，来规定该怎样解析和显示日期或时间的格式，默认的字符串是 ISO-8601 所给的日期时间组合格式“yyyy-MM-ddTHH:mm:ss”。数据信息部分表达日期的字符串必须符合声明中规定的格式要求（下文有例子）。

数据信息

数据信息中“@data”标记独占一行，剩下的是各个实例的数据。每个实例占一行。实例的各属性值用逗号“,”隔开。如果某个属性的值是缺失值（missing value），用问号“?”表示，且这个问号不能省略。例如：

@data
sunny,85,85,FALSE,no
?,78,90,?,yes

字符串属性和分类属性的值是区分大小写的。若值中含有空格，必须被引号括起来。例如：

@relation LCCvsLCSH
@attribute LCC string
@attribute LCSH string
@data
AG5, 'Encyclopedias and dictionaries.;Twentieth century.'
AS262, 'Science -- Soviet Union -- History.'

日期属性的值必须与属性声明中给定的相一致。例如：

@RELATION Timestamps
@ATTRIBUTE timestamp DATE "yyyy-MM-dd HH:mm:ss"
@DATA
"2001-04-03 12:12:12"
"2001-05-03 12:59:55"

稀疏数据

有的时候数据集中含有大量的 0 值（比如购物篮分析），这个时候用稀疏格式的数据存贮更加省空间。稀疏格式是针对数据信息中某个实例的表示而言，不需要修改 ARFF 文件的其它部分。看如下的数据：

@data

0, X, 0, Y, "class A"

0, 0, W, 0, "class B"

用稀疏格式表达的话就是

@data

{1 X, 3 Y, 4 "class A"}

{2 W, 4 "class B"}

每个实例用花括号括起来。实例中每一个非 0 的属性值用<index> <空格> <value>表示。<index>是属性的序号，从 0 开始计；<value>是属性值。属性值之间仍用逗号隔开。

注意在稀疏格式中没有注明的属性值不是缺失值，而是 0 值。若要表示缺失值必须显式的用问号表示出来。

Relational 型属性

在 WEKA 3.5 版中增加了一种属性类型叫做 Relational，有了这种类型我们可以像关系型数据库那样处理多个维度了。但是这种类型目前还不见广泛应用，暂不作介绍。

6.3. 数据准备

使用 WEKA 作数据挖掘，面临的第一个问题往往是我们的数据不是 ARFF 格式的。幸好，WEKA 还提供了对 CSV 文件的支持，而这种格式是被很多其他软件所支持的。此外，WEKA 还提供了通过 JDBC 访问数据库的功能。

在这一节里，我们先以 Excel 和 Matlab 为例，说明如何获得 CSV 文件。然后我们将知道 CSV 文件如何转化成 ARFF 文件，毕竟后者才是 WEKA 支持得最好的文件格式。面对一个 ARFF 文件，我们仍有一些预处理要做，才能进行挖掘任务。

.* -> .csv

我们给出一个 CSV 文件的例子（bank-data.csv）。用 UltraEdit 打开它可以看到，这种格式也是一种逗号分割数据的文本文件,储存了一个二维表格。

Excel 的 XLS 文件可以让多个二维表格放到不同的工作表（Sheet）中，我们只能把每个工作表存成不同的 CSV 文件。打开一个 XLS 文件并切换到需要转换的工作表，另存为 CSV 类型，点“确定”、“是”忽略提示即可完成操作。

在 Matlab 中的二维表格是一个矩阵，我们通过这条命令把一个矩阵存成 CSV 格式。

csvwrite('filename',matrixname)

需要注意的是，Matllab 给出的 CSV 文件往往没有属性名（Excel 给出的也有可能没有）。而 WEKA 必须从 CSV 文件的第一行读取属性名，否则就 会把第一行的各属性值读成变量名。

因此我们对于 Matllab 给出的 CSV 文件需要用 UltraEdit 打开，手工添加一行属性名。注意属性名的个数要跟 数据属性的个数一致，仍用逗号隔开。

.csv -> .arff

将 CSV 转换为 ARFF 最迅捷的办法是使用 WEKA 所带的命令行工具。

运行 WEKA 的主程序，出现 GUI 后可以点击下方按钮进入相应的模块。我们点击进入“Simple CLI”模块提供的命令行功能。在新窗口的最下方（上方是不能写字的）输入框写上

java weka.core.converters.CSVLoader filename.csv > filename.arff

即可完成转换。

在 WEKA 3.5 中提供了一个“Arff Viewer”模块，我们可以用它打开一个 CSV 文件将进行浏览，然后另存为 ARFF 文件。

进入“Exploer”模块，从上方的按钮中打开 CSV 文件然后另存为 ARFF 文件亦可。

“Exploer”界面

我们应该注意到，“Exploer”还提供了很多功能，实际上可以说这是 WEKA 使用最多的模块。现在我们先来熟悉它的界面，然后利用它对数据进行预处理。

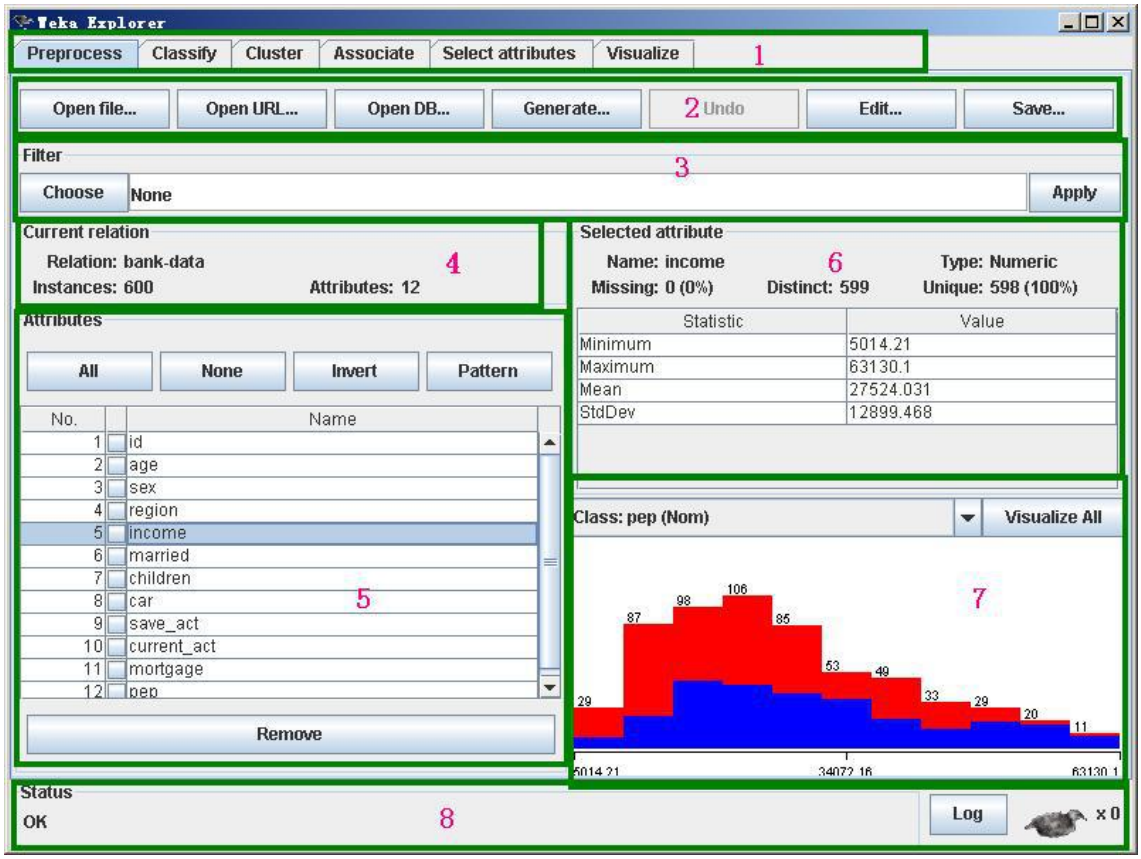

图 2 新窗口打开

图 2 显示的是使用 3.5 版"Exploer"打开"bank-data.csv"的情况。我们根据不同的功能把这个界

面分成 8 个区域。

区域 1 的几个选项卡是用来切换不同的挖掘任务面板。这一节用到的只有“Preprocess”，其他面板的功能将在以后介绍。

区域 2 是一些常用按钮。包括打开数据，保存及编辑功能。我们在这里把"bank-data.csv"另存为"bank-data.arff"。

在区域 3 中“Choose”某个“Filter”，可以实现筛选数据或者对数据进行某种变换。数据预处理主要就利用它来实现。

区域 4 展示了数据集的一些基本情况。

区域 5 中列出了数据集的所有属性。勾选一些属性并“Remove”就可以删除它们，删除后还可以利用区域 2 的“Undo”按钮找回。区域 5 上方的一排按钮是用来实现快速勾选的。

在区域 5 中选中某个属性，则区域 6 中有关于这个属性的摘要。注意对于数值属性和分类属性，摘要的方式是不一样的。图中显示的是对数值属性“income”的摘要。

区域 7 是区域 5 中选中属性的直方图。若数据集的最后一个属性（我们说过这是分类或回归任务的默认目标变量）是分类变量（这里的“pep”正好是），直方图 中的每个长方形就会按照该变量的比例分成不同颜色的段。要想换个分段的依据，在区域 7 上方的下拉框中选个不同的分类属性就可以了。下拉框里选上 “No Class”或者一个数值属性会变成黑白的直方图。

区域 8 是状态栏，可以查看 Log 以判断是否有错。右边的 weka 鸟在动的话说明 WEKA 正在执行挖掘任务。右键点击状态栏还可以执行 JAVA 内存的垃圾回收。

预处理

bank-data 数据各属性的含义如下：

id a unique identification number

age age of customer in years (numeric)

sex MALE / FEMALE

region inner_city/rural/suburban/town

income income of customer (numeric)

married is the customer married (YES/NO)

children number of children (numeric)

car does the customer own a car (YES/NO)

save_acct does the customer have a saving account (YES/NO)

current_acct does the customer have a current account (YES/NO)

mortgage does the customer have a mortgage (YES/NO)

pep did the customer buy a PEP (Personal Equity Plan) after the last mailing (YES/NO)

通常对于数据挖掘任务来说，ID 这样的信息是无用的，我们将之删除。在区域 5 勾选属性“id”，并点击“Remove”。将新的数据集保存一次，并用 UltraEdit 打开这个 ARFF 文件。我们发现，

在属性声明部分，WEKA 已经为每个属性选好了合适的类型。

我们知道，有些算法，只能处理所有的属性都是分类型的情况。这时候我们就需要对数值型的属性进行离散化。在这个数据集中有 3 个变量是数值型的，分别是“age”，“income”和“children”。

其中“children”只有 4 个取值：0，1，2，3。这时我们在 UltraEdit 中直接修改 ARFF 文件，把

@attribute children numeric

改为

@attribute children {0,1,2,3}

就可以了。

在“Explorer”中重新打开“bank-data.arff”，看看选中“children”属性后，区域 6 那里显示的“Type”是不是变成“Nominal”了

“age”和“income”的离散化我们需要借助 WEKA 中名为“Discretize”的 Filter 来完成。在区域 2 中点“Choose”，出现一棵“Filter 树”，逐级找到“weka.filters.unsupervised.attribute.Discretize”，点击。若无法关闭这个树，在树之外的地方点击“Explorer”面板即可。

现在“Choose”旁边的文本框应该显示“Discretize -B 10 -M -0.1 -R first-last”。点击这个文本框会弹出新窗口以修改离散化的参数。

我们不打算对所有的属性离散化，只是针对对第 1 个和第 4 个属性（见区域 5 属性名左边的数字），故把 attributeIndices 右边改成“1,4”。计划把这两个属性都分成 3 段，于是把“bins”改成“3”。其它框里不用更改，关于它们的意思可以点“More”查看。点“OK”回到“Explorer”，可以看到“age”和“income”已经被离散化成分类型的属性。若想放弃离散化可以点区域 2 的“Undo”。

如果对“"(-inf-34.333333]"”这样晦涩的标识不满，我们可以用 UltraEdit 打开保存后的 ARFF 文件，把所有的“\'(-inf-34.333333]\"”替换成“0_34”。其它标识做类似地手动替换。

经过上述操作得到的数据集我们保存为 bank-data-final.arff。

6.4. 关联规则（购物篮分析）

注意：目前，WEKA 的关联规则分析功能仅能用来作示范，不适合用来挖掘大型数据集。

我们打算对前面的“bank-data”数据作关联规则的分析。用“Explorer”打开“bank-data-final.arff”后，切换到“Associate”选项卡。默认关联规则分析是用 Apriori 算法，我们就用这个算法，但是点“Choose”右边的文本框修改默认的参数，弹出的窗口中点“More”可以看到各参数的说明。

背景知识

首先我们来温习一下 Apriori 的有关知识。对于一条关联规则 L->R，我们常用支持度（Support）和置信度（Confidence）来衡 量它的重要性。规则的支持度是用来估计在一个购物篮中同时观察到 L 和 R 的概率 P(L,R)，而规则的置信度是估计购物栏中出现了 L 时也出会现 R 的条件概率 P(R|L)。关联规则的目标一般是产生支持度和置信度都较高的规则。

有几个类似的度量代替置信度来衡量规则的关联程度，它们分别是

Lift（提升度？）： P(L,R)/(P(L)P(R))

Lift=1 时表示 L 和 R 独立。这个数越大，越表明 L 和 R 存在在一个购物篮中不是偶然现象。

Leverage（不知道怎么翻译）：P(L,R)-P(L)P(R)

它和 Lift 的含义差不多。Leverage=0 时 L 和 R 独立，Leverage 越大 L 和 R 的关系越密切。

Conviction（更不知道译了）：P(L)P(!R)/P(L,!R) （!R 表示 R 没有发生）

Conviction 也是用来衡量 L 和 R 的独立性。从它和 lift 的关系（对 R 取反，代入 Lift 公式后求倒数）可以看出，我们也希望这个值越大越好。

值得注意的是，用 Lift 和 Leverage 作标准时，L 和 R 是对称的，Confidence 和 Conviction 则不然。

参数设置

现在我们计划挖掘出支持度在 10%到 100%之间，并且 lift 值超过 1.5 且 lift 值排在前 100 位的那些关联规则。我们把 “lowerBoundMinSupport”和“upperBoundMinSupport”分别设为 0.1 和 1，“metricType”设为 lift，“minMetric”设为 1.5，“numRules”设为 100。其他选项保持默认即可。“OK” 之后在“Explorer”中点击 “Start”开始运行算法，在右边窗口显示数据集摘要和挖掘结果。

下面是挖掘出来的 lift 排前 5 的规则。

Best rules found:

1. age=52_max save_act=YES current_act=YES 113 ==> income=43759_max 61 conf:(0.54) < lift:(4.05)> lev:(0.0500)this.width=500" border=0> [45] conv:(1.85)

2. income=43759_max 80 ==> age=52_max save_act=YES current_act=YES 61 conf:(0.76) < lift:(4.05)> lev:(0.0500)this.width=500" border=0> [45] conv:(3.25) http://www.chinakdd.com

3. income=43759_max current_act=YES 63 ==> age=52_max save_act=YES 61 conf:(0.97) < lift:(3.85)> lev:(0.0500)this.width=500" border=0> [45] conv:(15.72)

4. age=52_max save_act=YES 151 ==> income=43759_max current_act=YES 61 conf:(0.4) < lift:(3.85)> lev:(0.0500)this.width=500" border=0> [45] conv:(1.49)

5. age=52_max save_act=YES 151 ==> income=43759_max 76 conf:(0.5) < lift:(3.77)> lev:(0.09) [55] conv:(1.72)

对于挖掘出的每条规则，WEKA 列出了它们关联程度的四项指标。

命令行方式

我们也可以利用命令行来完成挖掘任务，在“Simlpe CLI”模块中输入如下格式的命令：

java weka.associations.Apriori options -t directory-path\bank-data-final.arff

即可完成 Apriori 算法。注意，“-t”参数后的文件路径中不能含有空格。

在前面我们使用的 option 为

-N 100 -T 1 -C 1.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 命令行中使用这些参数得到的结果和前面利用 GUI 得到的一样。

我们还可以加上“- I”参数，得到不同项数的频繁项集。我用的命令如下：

java weka.associations.Apriori -N 100 -T 1 -C 1.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -I -t d:\weka\bank-data-final.arff

挖掘结果在上方显示，应是这个文件的样子。

6.5. 分类与回归

背景知识

WEKA 把分类(Classification)和回归(Regression)都放在“Classify”选项卡中，这是有原因的。

在这两个任务中，都有一个目标属性（输出变量）。我们希望根据一个样本(WEKA 中称作实例)的一组特征（输入变量），对目标进行预测。为了实现这一目 的，我们需要有一个训练数据集，这个数据集中每个实例的输入和输出都是已知的。观察训练集中的实例，可以建立起预测的模型。有了这个模型，我们就可以新的 输出未知的实例进行预测了。衡量模型的好坏就在于预测的准确程度。

在 WEKA 中，待预测的目标（输出）被称作 Class 属性，这应该是来自分类任务的“类”。一般的，若 Class 属性是分类型时我们的任务才叫分类，Class 属性是数值型时我们的任务叫回归。

选择算法

这一节中，我们使用 C4.5 决策树算法对 bank-data 建立起分类模型。

我们来看原来的“bank-data.csv” 文件。“ID”属性肯定是不需要的。由于 C4.5 算法可以处理数值型的属性，我们不用像前面用关联规则那样把每个变量都离散化成分类型。尽管如此，我们还 是把“Children”属性转换成分类型的两个值“YES”和“NO”。另外，我们的训练集仅取原来数据集实例的一半；而从另外一半中抽出若干条作为待 预测的实例，它们的“pep”属性都设为缺失值。经过了这些处理的训练集数据在这里下载；待预测集数据在这里下载。

我们用“Explorer”打开训练集“bank.arff”，观察一下它是不是按照前面的要求处理好了。切换到“Classify”选项卡，点击 “Choose”按钮后可以看到很多分类或者回归的算法分门别类的列在一个树型框里。3.5 版的 WEKA 中，树型框下方有一个“Filter...”按 钮，点击可以根据数据集的特性过滤掉不合适的算法。我们数据集的输入属性中有“Binary”型（即只有两个类的分类型）和数值型的属性，而 Class 变 量是“Binary”的；于是我们勾选“Binary

attributes”“Numeric attributes”和 “Binary class”。点“OK”后回到树形图，可以发现一些算法名称变红了，说明它们不能用。选择“trees”下的“J48”，这就是我们需 要的 C4.5 算法，还好它没有变红。

点击“Choose”右边的文本框，弹出新窗口为该算法设置各种参数。点“More”查看参数说明，点“Capabilities”是查看算法适用范围。这里我们把参数保持默认。

现在来看左中的“Test Option”。我们没有专门设置检验数据集，为了保证生成的模型的准确性而不至于出现过拟合（overfitting）的现 象，我们有必要采用 10 折交叉验证（10-fold cross validation）来选择和评估模型。若不明白交叉验证的含义可以 Google 一下。

建模结果

OK，选上“Cross-validation”并在“Folds”框填上“10”。点“Start”按钮开始让算法生成决策树模型。很快，用文本表示的 一棵决策树，以及对这个决策树的误差分析等等结果出现在右边的“Classifier output”中。同时左下的“Results list”出现了 一个项目显示刚才的时间和算法名称。如果换一个模型或者换个参数，重新“Start”一次，则“Results list”又会多出一项。

我们看到“J48”算法交叉验证的结果之一为

Correctly Classified Instances 206 68.6667 %

也就是说这个模型的准确度只有 69%左右。也许我们需要对原属性进行处理，或者修改算法的参数来提高准确度。但这里我们不管它，继续用这个模型。

右键点击“Results list”刚才出现的那一项，弹出菜单中选择“Visualize tree”，新窗口里可以看到图形模式的决策树。建议把这个新窗口最大化，然后点右键，选“Fit to screen”，可以把这个树看清楚些。看完后截图或者关掉。

这里我们解释一下“Confusion Matrix”的含义。

```
=== Confusion Matrix ===
  a b <-- classified as
  74 64 | a = YES
  30 132 | b = NO
```

这个矩阵是说，原本“pep”是“YES”的实例，有 74 个被正确的预测为“YES”，有 64 个错误的预测成了“NO”；原本“pep”是“NO”的实 例，有 30 个被错误的预测为“YES”，有 132 个正确的预测成了“NO”。74+64+30+132 = 300 是实例总数，而 (74+132)/300 = 0.68667 正好是正确分类的实例所占比例。这个矩阵对角线上的数字越大，说明预测得越好。

模型应用

现在我们要用生成的模型对那些待预测的数据集进行预测了，注意待预测数据集和训练用数据集各个属性的设置必须是一致的。WEKA 中并没有直接提供把模型应用到带预测数据集上的方法，我们要采取间接的办法。

在“Test Opion”中选择“Supplied test set”，并且“Set”成“bank-new.arff”文件。

现在，右键点击“Result list”中刚产生的那一项，选择“Visualize classifier errors”。我

们不去管新窗口中的图 有什么含义，点“Save”按钮，把结果保存成“bank-predicted.arff”。这个 ARFF 文件中就有我们需要的预测结果。在 “Explorer”的“Preprocess”选项卡中打开这个新文件，可以看到多了两个属性“Instance_number”和 “predictedpep”。“Instance_number”是指一个实例在原“bank-new.arff”文件中的位 置，“predictedpep”就是模型预测的结果。点“Edit”按钮或者在“ArffViewer”模块中打开可以查看这个数据集的内容。比如，我 们对实例 0 的 pep 预测值为“YES”，对实例 4 的预测值为“NO”。

使用命令行（推荐）

虽然使用图形界面查看结果和设置参数很方便，但是最直接最灵活的建模及应用的办法仍是使用命令行。

打开“Simple CLI”模块，像上面那样使用“J48”算法的命令格式为：

java weka.classifiers.trees.J48 -C 0.25 -M 2 -t directory-path\bank.arff -d directory-path \bank.model

其中参数“ -C 0.25”和“-M 2”是和图形界面中所设的一样的。“-t ”后面跟着的是训练数据集的完整路径（包括目录和文件名），“-d ”后面跟着的是保存模型的完整路径。注意！这里我们可以把模型保存下来。

输入上述命令后，所得到树模型和误差分析会在“Simple CLI”上方显示，可以复制下来保存在文本文件里。误差是把模型应用到训练集上给出的。

把这个模型应用到“bank-new.arff”所用命令的格式为：

java weka.classifiers.trees.J48 -p 9 -l directory-path\bank.model -T directory-path \bank-new.arff

其中“-p 9”说的是模型中的 Class 属性是第 9 个（也就是“pep”），“-l”后面是模型的完整路径，“-T”后面是待预测数据集的完整路径。

输入上述命令后，在“Simple CLI”上方会有这样一些结果：

0 YES 0.75 ?

1 NO 0.7272727272727273 ?

2 YES 0.95 ?

3 YES 0.8813559322033898 ?

4 NO 0.8421052631578947 ?

...

这里的第一列就是我们提到过的“Instance_number”，第二列就是刚才的“predictedpep”，第四列则是“bank- new.arff”中原来的“pep”值（这里都是“?”缺失值）。第三列对预测结果的置信度（confidence ）。比如说对于实例 0，我们有 75%的把握说它的“pep”的值会是“YES”，对实例 4 我们有 84.2%的把握说它的“pep”值会是“NO”。

我们看到，使用命令行至少有两个好处。一个是可以把模型保存下来，这样有新的待预测数据出现时，不用每次重新建模，直接应用保存好的模型即可。另一个是对预测结果给出了置信度，我们可以有选择的采纳预测结果，例如，只考虑那些置信度在 85%以上的结果。可惜，命令行仍不能保存交叉验证等方式选择过的模型，也不能将它们应用到待预测数据上。

要实现这一目的，须用到“KnowledgeFlow”模块的“ PredictionAppender”。

6.6. 聚类分析

原理与实现

聚类分析中的“类”（cluster）和前面分类的“类”（class）是不同的，对 cluster 更加准确的翻译应该是“簇”。聚类的任务是把所有的实 例分配到若干的簇，使得同一个簇的实例聚集在一个簇中心的周围，它们之间距离的比较近；而不同簇实例之间的距离比较远。对于由数值型属性刻画的实例来说， 这个距离通常指欧氏距离。

现在我们对前面的“bank data”作聚类分析，使用最常见的 K 均值（K-means）算法。下面我们简单描述一下 K 均值聚类的步骤。

K 均值算法首先随机的指定 K 个簇中心。然后：1)将每个实例分配到距它最近的簇中心，得到 K 个簇；2)计分别计算各簇中所有实例的均值，把它们作为各簇新的簇中心。重复 1)和 2)，直到 K 个簇中心的位置都固定，簇的分配也固定。

上述 K 均值算法只能处理数值型的属性，遇到分类型的属性时要把它变为若干个取值 0 和 1 的属性。WEKA 将自动实施这个分类型到数值型的变换，而且 WEKA 会自动对数值型的数据作标准化。因此，对于原始数据“bank-data.csv”，我们所做的预处理只是删去属性“id”，保存为 ARFF 格式后，修改 属性“children”为分类型。这样得到的数据文件为“bank.arff”，含 600 条实例。

用“Explorer”打开刚才得到的“bank.arff”，并切换到“Cluster”。点“Choose”按钮选择“SimpleKMeans”， 这是 WEKA 中实现 K 均值的算法。点击旁边的文本框，修改“numClusters”为 6，说明我们希望把这 600 条实例聚成 6 类，即 K=6。下面的 “seed”参数是要设置一个随机种子，依此产生一个随机数，用来得到 K 均值算法中第一次给出的 K 个簇中心的位置。我们不妨暂时让它就为 10。

选中“Cluster Mode”的“Use training set”，点击“Start”按钮，观察右边“Clusterer output”给出 的聚类结果。也可以在左下角“Result list”中这次产生的结果上点右键，“View in separate window”在新窗口中浏览结 果。

结果解释

首先我们注意到结果中有这么一行：

Within cluster sum of squared errors: 1604.7416693522332

这是评价聚类好坏的标准，数值越小说明同一簇实例之间的距离越小。也许你得到的数值会不一样；实际上如果把“seed”参数改一下，得到的这个数值就可能会不一样。我们应该多尝试几个 seed，并采纳这个数值最小的那个结果。例如我让“seed”取 100，就得到

Within cluster sum of squared errors: 1555.6241507629218

我该取后面这个。当然再尝试几个 seed，这个数值可能会更小。

接下来“Cluster centroids:”之后列出了各个簇中心的位置。对于数值型的属性，簇中心就是它的均值（Mean）；分类型的就是它的众数 （Mode）， 也就是说这个属性上取值为众数值的实例最多。对于数值型的属性，还给出了它在各个簇里的标准差（Std Devs）。

最后的“Clustered Instances”是各个簇中实例的数目及百分比。

为了观察可视化的聚类结果，我们在左下方“Result list”列出的结果上右击，点“Visualize cluster assignments”。弹出的窗口给出了各实例的散点图。最上方的两个框是选择横坐标和纵坐标，第二行的 “color”是散点图着色的依据，默认是根据不同的簇“Cluster”给实例标上不同的颜色。

可以在这里点“Save”把聚类结果保存成 ARFF 文件。在这个新的 ARFF 文件中，“instance_number”属性表示某实例的编号，“Cluster”属性表示聚类算法给出的该实例所在的簇。