云南大学数学与统计学院  
《数据挖掘与决策支持实验》上机实践报告

|  |  |  |
| --- | --- | --- |
| **课程名称**：运筹学数据挖掘与决策支持实验 | **年级**：2015级 | **上机实践成绩**： |
| **指导教师**：彭程 | **姓名**：刘鹏 | **专业：**信息与计算科学 |
| **上机实践名称**：对机器生产数据进行特征选择 | **学号**：20151910042 | **上机实践日期**：2018-07-04 |
| **上机实践编号**：01 | **组号**： |  |

# 一、实验目的

学习使用R语言进行变量选择。

# 二、实验内容

如下表：

表格 1 产品加工与产品良率

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 产品编号 | 加工时间/h | 机台类型 | 加工时间/h | 机台类型 | 良率 |
| 1 | 28 | A01 | 48 | B03 | 0.53 |
| 2 | 27 | A01 | 42 | B03 | 0.62 |
| 3 | 31 | A03 | 43 | B21 | 0.84 |
| 4 | 42 | A02 | 33 | B02 | 0.91 |
| 5 | 46 | A02 | 28 | B03 | 0.85 |
| 6 | 50 | A01 | 27 | B03 | 0.68 |
| 7 | 35 | A02 | 24 | B01 | 0.83 |
| 8 | 24 | A03 | 36 | B02 | 0.69 |
| 9 | 28 | A02 | 25 | B01 | 0.88 |
| 10 | 44 | A03 | 37 | B03 | 0.92 |

请将给定数据进行变量选择，从而实现维归约。要求使用一种基于熵度量的无监督特征选择方法减少数据集的维度。

# 三、实验平台

Windows 10 Pro 1803；

Microsoft© Visual Studio 2017 Enterprise。

Version 1.1.442 – © 2009-2018 RStudio, Inc.

# 四、算法设计

数据预分析：如表格 1 产品加工与产品良率所示，表格中有四列自变量，一列因变量。其中，因为每种产品都需要进行两个阶段的加工，所以有加工时间与对应的机台类型。可以观察到，加工时间是属于有顺序关系的数值型数据，而机台类型是属于分类型数据。

算法背景介绍：数据规约过程的三个基本操作是删除列，删除行，减少列中值的数量（平整特征）。这些操作试图删掉不必要的数据来保留原始数据的特征。减少维度还有其他操作，但是和原始数据集相比，新数据是未被认可的。数据的最终归约不会降低结果的质量，在某些应用中，数据挖掘结果甚至得到了改善。理想情况下，使用维归约既能减少时间，又能提高精度、简化模型的描述。

数据归约算法的推荐特性如下，它们是这些技术的设计者设计算法的指导方针[1] 。

1. 可测性 应用已归约的数据集可精确地确定近似结果的质量。
2. 可辨识 在应用数据挖掘程序之前，在数据归约算法运行期间，很容易确定近似结果的质量。
3. 单一性 算法往往是迭代的，计算结果的质量是时间和输入数据质量的一个非递减函数。
4. 一致性 计算结果的质量与计算时间以及输入数据质量有关。
5. 收益递减 方案在计算的早期（迭代）能获得大的改进，但随时间递减。
6. 可中断性 算法可以随时停止，并给出答案。
7. 优先权 算法可以暂停并以最小的开销重新开始。

在进行数据挖掘时，我们并不需要将所有的自变量用来建模，而是从所有的变量中选择最重要的变量，这称为变量选择（feature selection）。一种算法是向后选择，即将所有的变量都包括在模型中，再次计算效能，反复迭代，找出合适的自变量数目。

通常，如果一个变量描述了不同种类的实体，则可以检查不同种类的样本。用变量的方差对变量的均值进行标准化，然后比较不同种类的标准化值。如果标准化均值相差很大，说明这个特征就很重要，反之说明两者的互信息比较大，两者所含信息量重叠很多，可以去掉其中之一。这种思想代表的是一种试探性的、非优化的特征选择方法。不过这种方法符合很多将数据挖掘技术应用于特征分类的实际经验。下面是检验公式：

从样本数据中可以看到，良率全都在百分之五十之上，不妨取0.80作为阈值进行检测。

# 五、程序代码

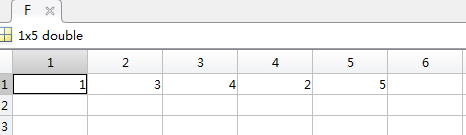
5.1 程序描述

5.2 程序代码

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | a **=** 30**;**  b **=** 40**;**  c **=** 0.8**;**  G **=** 4**;**  n **=** length**(**T**{**1**);**  m **=** length**(**T**);**  **for** i **=** 1**:**m  T**{**i**}** **=** cell2mat**(**T**{**i**});**  **end**  T**{**1**}** **=** **(**T**{**1**}** **>** a**)+**0**;**  T**{**2**}** **=** T**{**2**}(:,**3**);**  T**{**3**}** **=** **(**T**{**3**}** **>** b**)+**0**;**  T**{**4**}** **=** T**{**4**}(:,**3**);**  T**{**5**}** **=** **(**T**{**5**}** **>** c**)+**0**;**      **for** k **=** 1**:**m  **for** i **=** 1**:**n**-**1  **for** j **=** i**+**1**:**n  Y**(**i**,**j**,**k**)** **=** 0 **+** **(**T**{**k**}(**i**)** **==** T**{**k**}(**j**));**  **end**  **end**  **end**  S **=** sum**(**Y**,**3**)./**m**;**  E **=** entropy**(**S**);**  index **=** 1**:**m    **for** i **=** m**:-**1**:**2  **for** k **=** 1**:**i  Sf**(:,:,**k**)** **=** **(**i**.\***S**-**Y**(:,:,**k**))./(**i**-**1**);**  Ef**(**k**)** **=** E **-** entropy**(**Sf**(:,:,**k**));**  **end**    **if** min**(**Ef**)** **<** G  fk **=** find**(**Ef **==** min**(**Ef**));**  index**(**fk**(**1**))** **=** **[];**  F**(**i**)** **=** fk**(**1**);**  E **=** E **-** Ef**(**fk**(**1**));**  Ef**(**fk**(**1**))** **=** **[];**  S **=** Sf**(:,:,**fk**(**1**));**  **else**  **break;**  **end**  **end**  F**(**1**:**i**)** **=** index**(**1**:**i**);** |

程序代码 1

5.3 运行结果



# 六、实验体会

# 七、参考文献

[1] KANTARDZIC M. 数据挖掘：概念模型方法和算法 [M]. 2nd ed. 北京: 清华大学出版社, 2013.