超市数据的频繁项集挖掘 – c问分析文档

数据分析与数据挖掘

DAM COURSE, SPRING 2018

BY

1552674 李　源

Tongji University
School of Software Engineering

1 代码运行结果

1.1 基于频繁项集的预测结果

选择aii问的pluno字段输出得到的频繁项集，其中支持度前十的频繁项集作为预测，及其在后40%中的支持度：

```
Top 10:
[['30380003'], 230]
[['30380002'], 158]
[['23110009'], 110]
[['30380002', '30380003'], 105]
[['22036000'], 89]
[['22102014'], 62]
[['23110001'], 59]
[['27410000'], 57]
[['23110009', '30380003'], 53]
[['22102005'], 50]
Test result:
(([['30380003'], 230], 158, 0.32644628099173556)
(([['30380002'], 158], 96, 0.19834710743801653)
(([['23110009'], 110], 86, 0.17768595041322313)
(([['30380002', '30380003'], 105], 41, 0.08471074380165289)
(([['22036000'], 89], 63, 0.13016528925619836)
(([['22102014'], 62], 53, 0.10950413223140495)
(([['23110001'], 59], 38, 0.07851239669421488)
(([['27410000'], 57], 66, 0.13636363636363635)
(([['23110009', '30380003'], 53], 36, 0.0743801652892562)
(([['22102005'], 50], 42, 0.08677685950413223)
```

选择aii问的bndno字段输出得到的频繁项集，其中支持度前十的频繁项集作为预测，及其在后40%中的支持度：

```
Top 10:
[['30248'], 284]
[['15094'], 152]
[['15012'], 136]
[['15052'], 120]
[['15094', '30248'], 103]
[['15012', '30248'], 99]
[['15052', '30248'], 84]
[['15039'], 76]
[['15631'], 74]
[['15009'], 62]
Test result:
(([['30248'], 284], 219, 0.47300215982721383)
(([['15094'], 152], 120, 0.2591792656587473)
(([['15012'], 136], 111, 0.23974082073434125)
(([['15052'], 120], 94, 0.20302375809935205)
(([['15094', '30248'], 103], 66, 0.14254859611231102)
(([['15012', '30248'], 99], 59, 0.12742980561555076)
(([['15052', '30248'], 84], 53, 0.11447084233261338)
(([['15039'], 76], 68, 0.1468682505399568)
(([['15631'], 74], 68, 0.1468682505399568)
(([['15009'], 62], 66, 0.14254859611231102)
```

1.2 基于关联规则的预测结果

选择aii问的pluno字段输出得到的频繁项集，用来生成候选的关联规则，前60%中的置信度最高的前十关联规则作为预测，及其在后40%中的置信度:

```
Top 10 confidence:
((frozenset({'10150006'}), '30380003'), 1.0)
((frozenset({'15115034'}), '30380002'), 1.0)
((frozenset({'15115034', '30380003'}), '30380002'), 1.0)
((frozenset({'22020000', '30380002'}), '30380003'), 1.0)
((frozenset({'22102014', '22171000'}), '30380003'), 1.0)
((frozenset({'22102014', '22171000'}), '30380002'), 1.0)
((frozenset({'22102014', '22171000', '30380003'}), '30380002'), 1.0)
((frozenset({'22102014', '22171000', '30380002'}), '30380003'), 1.0)
((frozenset({'30380002', '22171000'}), '30380003'), 1.0)
((frozenset({'22170001', '22171000'}), '30380003'), 1.0)
Test result:
((frozenset({'22102014', '22171000'}), '30380003'), 1.0)
((frozenset({'22102014', '22171000'}), '30380002'), 1.0)
((frozenset({'22102014', '22171000', '30380003'}), '30380002'), 1.0)
((frozenset({'22102014', '22171000', '30380002'}), '30380003'), 1.0)
((frozenset({'30380002', '22171000'}), '30380003'), 1.0)
((frozenset({'22170001', '22171000'}), '30380003'), 1.0)
((frozenset({'15115034'}), '30380002'), 0.0)
((frozenset({'15115034', '30380003'}), '30380002'), 0.0)
```

选择aii问的pluno字段输出得到的频繁项集，用来生成候选的关联规则，前60%中的置信度最高的前十关联规则作为预测，及其在后40%中的置信度:

```
Top 10 confidence:
((frozenset({'30060'}), '30248'), 1.0)
((frozenset({'14396'}), '30248'), 1.0)
((frozenset({'14020'}), '30248'), 1.0)
((frozenset({'14737'}), '30248'), 1.0)
((frozenset({'14071'}), '30248'), 1.0)
((frozenset({'14322', '14071'}), '30248'), 1.0)
((frozenset({'15545'}), '30248'), 1.0)
((frozenset({'15545', '15094'}), '30248'), 1.0)
((frozenset({'10632'}), '30248'), 1.0)
((frozenset({'14509'}), '30248'), 1.0)
Test result:
((frozenset({'30060'}), '30248'), 1.0)
((frozenset({'14396'}), '30248'), 1.0)
((frozenset({'15545', '15094'}), '30248'), 1.0)
((frozenset({'14737'}), '30248'), 0.6666666666666666)
((frozenset({'15545'}), '30248'), 0.6666666666666666)
((frozenset({'14071'}), '30248'), 0.5)
((frozenset({'14509'}), '30248'), 0.5)
((frozenset({'14322', '14071'}), '30248'), 0.3333333333333333)
((frozenset({'10632'}), '30248'), 0.25)
((frozenset({'14020'}), '30248'), 0.0)
```

80%的关联规则都在后40%取得了很高的置信度，说明用这些关联规则进行预测效果较好。
其他的字段得到的效果是相近的，这里为了不做一一展示了。

2 分析讨论

这里主要分别对两种预测方法进行分析讨论。

2.1 基于频繁项集的预测

在a问和b问中，我已经得到了频繁项集的输出，那么我们可以直接考虑选择支持度较高的一些频繁项集，认为是用户接下来的购买信息。

我选择了选择a、b问的字段输出得到的频繁项集，其中支持度前十的频繁项集作为预测，然后在后40%的数据中，针对每一个用户进行评估，得到的结果如前文所示。

可以发现，前60%中的频繁项集在后40%中数据的支持度也比较高，并且其对排序也是相近的。那么我们可以认为，直接用生成的频繁项集来预测用户的购买信息效果是比较好的。也可以这样描述，前60%中的频繁项集是适用于后40%的数据。

2.2 基于关联规则的预测

当然，我们可以进一步通过生成关联规则来对用户的购买信息进行预测。

我们可以从频繁项集中抽取出关联规则，把几个购买信息作为作为前提，剩下的一个购买信息作为结论组成如下形式的规则：**如果用户购买了前提中的所有商品，那么他们也会想购买结论中的商品**。每一条频繁项集都可以生成几条这样的候选关联规则。

接下来，计算每条规则的置信度，这里的计算方法大致如下：

1）先创建两个字典，用来存储规则应验（正例）和规则不适用（反例）的次数。

2）遍历所有用户的购买信息，在这个过程中遍历每条关联规则。

3）逐个计算是否应验。

4）用规则应验的次数除以前提条件出现的总次数，计算每条规则的置信度。

5）排序选择置信度前十的关联规则作为预测方法。

6）在后40%的数据上评估发现的规则在测试集上的表现。具体的输出在前文有描述。

可以发现，前60%的数据得到的关联规则，在后40%中也得到了很好的置信度。可以这样认为，利用关联规则来进行用户的预测，也能够取得一个很好的结果。

同时，我将trade.csv和trade_new.csv都进行了预测，发现得到的效果是相近的。