

商品关联分析篇

商业智能部数据挖掘课题



2015-12-31

数字科技有限公司

广州市广州大道南敦和路 189号海珠科技园3号楼10层

目录

[课题介绍 1](#_Toc28870)

[一、前言 1](#_Toc31058)

[二、成果 1](#_Toc25861)

[三、其他 1](#_Toc10394)

[商品关联说明书 2](#_Toc17856)

[一、关联原理 2](#_Toc719)

[1、支持度(support) 2](#_Toc27024)

[2、置信度(confidence) 2](#_Toc2350)

[3、期望置信度(Expected confidence) 3](#_Toc26018)

[4、提升度(lift) 3](#_Toc14733)

[5、总结 3](#_Toc28770)

[二、关联算法 4](#_Toc10452)

[1、Apriori算法 4](#_Toc19103)

[算法介绍 4](#_Toc7691)

[术语说明 4](#_Toc27342)

[算法步骤 5](#_Toc6942)

[总结 6](#_Toc17231)

[2、FP-tree算法 6](#_Toc26763)

[算法介绍 6](#_Toc990)

[算法步骤 6](#_Toc9094)

[总结 10](#_Toc30618)

[3、集群计算 10](#_Toc2663)

[三、算法实现 10](#_Toc27727)

[1、运行环境 10](#_Toc24924)

[2、输入数据 11](#_Toc13528)

[3、程序处理 14](#_Toc22952)

[TreeNode.java 15](#_Toc10204)

[Mysql.java 15](#_Toc30723)

[FPTree.java 15](#_Toc17718)

[4、输出数据 15](#_Toc6869)

[四、结果分析 16](#_Toc16320)

[五、参考 18](#_Toc1885)

[达芙妮官方旗舰店 商品关联案例分析 19](#_Toc22108)

[一、 店铺介绍 19](#_Toc23817)

[二、 经营商品 19](#_Toc32374)

[三、 数据情况 21](#_Toc2180)

[四、 结果分析 21](#_Toc13133)

[五、需求更改后 26](#_Toc23624)

# 课题介绍

## 一、前言

公司作为电子商务运营商，每天需要与成千上万的在线商品打交道，并因此积累了大量线上订单数据。这些**历史数据堆积在数据库中是毫无用处的**，通过数据挖掘，或许可以得到一些有用的信息，从而对公司业务进行一定的指导，提高各在线店铺的营业额。

**啤酒和尿布**的故事人尽皆知，购买啤酒的大部分人同时也购买了尿布，于是通过组合啤酒和尿布进行促销，人们更方便地购买了两者，同时销量额大幅提高。本课题的原理亦相似。

## 二、成果

本课题要达到的成果如下：

1. 根据数学建模，完成详尽的**商品关联分析说明书**，详细到课题的方方面面，以此作为指导核心。最终产物是文档形式。此工作由商业智能部的数学建模小组负责。
2. 根据商品分析说明书，完成商品关联或推荐模块需求分析，设计分析，必要时进行原型设计，并进行编码实现，整合进公司内部**CRM系统**，实现线上可视化，方便非技术人员使用，此工作由技术部负责。

补充：商品关联开发周期请视具体情况自行安排，任何疑问可联系商业智能数学建模小组。

## 三、其他

商品关联之前需要各店铺对自己的产品进行**秩序化类目编号**，由各店铺负责人负责。一个要经营长久的店铺，其商品一定要有自己的分类系统，分类混乱的店铺势必在发货，记账等环节造成不必要的麻烦。编号可参考图书ISBN。

ISBN EAN•UCC前缀-组区号-出版者号-出版序号-校验码

示例：ISBN 978-7-5064-2595-7

中国标准书号数字的第一部分。由国际物品编码（EAN·UCC）系统专门提供给国际ISBN管理系统的产品标识编码。 中国标准书号数字的第二部分。它由国际ISBN管理机构分配。中国的组区号为“7”。中国标准书号数字的第三部分。标识具体的出版者。其长度为2至7位，由中国ISBN管理机构设置和分配。中国标准书号数字的第四部分。由出版者按出版物的出版次序管理和编制。中国标准书号数字的第五部分，也是其最后一位。采用模数10加权算法计算得出。

# 商品关联说明书

## 一、关联原理

关联规则挖掘属于**[无监督学习方法](http://blog.sina.com.cn/s/blog_8813a3ae01018dan.html" \t "_blank)**，它描述的是在**一个事物中物品间同时出现的规律的知识模式**，现实生活中，比如超市购物时，顾客购买记录常常隐含着很多关联规则，比如购买圆珠笔的顾客中有65%也购买了笔记本，利用这些规则，商场人员可以很好的**规划商品摆放问题**。

为叙述方便，设R= {I1,I2 ......Im} 是一组物品集，W是一组事务集。W中的每个事务T是一组物品，T是R的子集。

假设有一个物品集A，一个事务T，关联规则是如下形式的一种蕴含：A→B，其中A、B是两组物品，A属于I子集，B属于I子集。

在关联规则中设计4个常用关键指标如下。

### 1、支持度(support)

**定义：**设W中有s％的事务同时支持物品集A和B，s％称为关联规则A→B的支持度。支持度描述了A和B这两个物品集的并集C在所有的事务中出现的概率有多大。

**通俗解释：**简单地说，A==>B的支持度就是指物品集A和物品集B同时出现的概率。

**实例说明：某天共有1000 个顾客到商场购买物品，其中有150个顾客同时购买了圆珠笔和笔记本，那么上述的关联规则的支持度就是15％。**

**概率描述：**物品集A对物品集B的支持度support(A==>B)=P(A n B)

### 2、置信度(confidence)

**定义：**设W中支持物品集A的事务中，有c％的事务同时也支持物品集B，c％称为关联规则A→B 的可信度。

**通俗解释：**简单地说，可信度就是指在出现了物品集A的事务T中，物品集B也同时出现的概率有多大。

**实例说明：上面所举的圆珠笔和笔记本的例子，该关联规则的可信度就回答了这样一个问题：如果一个顾客购买了圆珠笔，那么他也购买笔记本的可能性有多大呢？在上述例子中，购买圆珠笔的顾客中有65％的人购买了笔记本,所以可信度是65％。**

**概率描述：**物品集A对物品集B的置信度confidence(A==>B)=P(A|B)

### 3、期望置信度(Expected confidence)

**定义：**设W中有e％的事务支持物品集B，e％称为关联规则A→B的期望可信度度。

**通俗解释：**期望可信度描述了在没有任何条件影响时，物品集B在所有事务中出现的概率有多大。

**实例说明：如果某天共有1000个顾客到商场购买物品，其中有250个顾客购买了圆珠笔，则上述的关联规则的期望可信度就是25％。**

**概率描述：**物品集A对物品集B的期望置信度为Expected confidence(B)=P(B)。相反，物品集B对物品集A的期望置信度为Expected confidence(A)=P(A)。

### 4、提升度(lift)

**定义：**提升度是可信度与期望可信度的比值

**通俗解释：**提升度反映了“物品集A的出现”对物品集B的出现概率发生了多大的变化。

**实例说明：**上述的关联规则的提升度=65%/25%=2.6

**概率描述：**物品集A对物品集B的提升度为

lift(A==>B)=confidence(A==>B)/ Expected confidence (B)=p(B|A)/p(B)

注意此条公式也可化为 P(A n B)/P(A)P(B)

### 5、总结

**可信度是对关联规则的[准确度](http://blog.sina.com.cn/u/2282988462" \t "_blank)的衡量，**[**支持度**](http://blog.sina.com.cn/u/2282988462)**是对关联规则重要性的衡量。**

支持度说明了这条规则在所有事务中有多大的代表性，显然支持度越大，关联规则越重要。有些关联规则可信度虽然很高，但支持度却很低，说明该关联规则实用的机会很小，因此也不重要。

总之，支持度是第一重要的，决定是否有必要进行该组合研究。置信度是第二重要的，决定组合是否真的强关联还是仅仅因为购买基数大。

在关联规则挖掘中，**满足一定最小置信度以及支持度的集合成为频繁集(frequent itemset)，或者强关联。**关联规则挖掘则是一个寻找频繁集的过程。

## 二、关联算法

### 1、Apriori算法

#### 算法介绍

**Apriori算法（先验算法）**是一种最有影响的挖掘布尔关联规则频繁项集的算法。其核心是基于两阶段频集思想的递推算法。该关联规则在分类上属于单维、单层、布尔关联规则。在这里，所有支持度大于最小支持度的项集称为频繁项集，简称频集。

该算法的基本思想是：首先找出所有的频集，这些项集出现的频繁性至少和预定义的最小支持度一样。然后由频集产生强关联规则，这些规则必须满足最小支持度和最小可信度。然后使用第1步找到的频集产生期望的规则，产生只包含集合的项的所有规则，其中每一条规则的右部只有一项，这里采用的是中规则的定义。一旦这些规则被生成，那么只有那些大于用户给定的最小可信度的规则才被留下来。为了生成所有频集，使用了递推的方法。

#### 术语说明

①支持度：P(A ∩B)，既有A又有B的概率

②置信度：P(B|A)，在A发生的事件中同时发生B的概率 p(AB)/P(A)

例如购物篮分析：牛奶 ⇒ 面包

例子：[支持度：3%，置信度：40%]

支持度3%：意味着3%顾客同时购买牛奶和面包

置信度40%：意味着购买牛奶的顾客40%也购买面包

③如果事件A中包含k个元素，那么称这个事件A为k项集事件A满足最小支持度阈值的事件称为频繁k项集。

④同时满足最小支持度阈值和最小置信度阈值的规则称为强规则

#### 算法步骤

1、依据支持度找出所有频繁项集（频度）

2、依据置信度产生关联规则（强度）。

Apriori使用一种称作逐层搜索的迭代方法，“K-1项集”用于搜索“K项集”。首先，找出频繁“1项集”的集合，该集合记作L1。L1用于找频繁“2项集”的集合L2，而L2用于找L3。如此下去，直到不能找到“K项集”。找每个Lk都需要一次数据库扫描。

核心思想是：连接步和剪枝步。连接步是自连接，原则是保证前k-2项相同，并按照字典顺序连接。剪枝步，是使任一频繁项集的所有非空子集也必须是频繁的。反之，如果某个候选的非空子集不是频繁的，那么该候选肯定不是频繁的，从而可以将其从CK中删除。

简单的讲，1、发现频繁项集，过程为（1）扫描（2）计数（3）比较（4）产生频繁项集（5）连接、剪枝，产生候选项集，重复步骤（1）~（5）直到不能发现更大的频集。如下图。

最后根据生成的频繁项集产生关联规则，过程为：根据前面提到的置信度的定义，关联规则的产生如下：

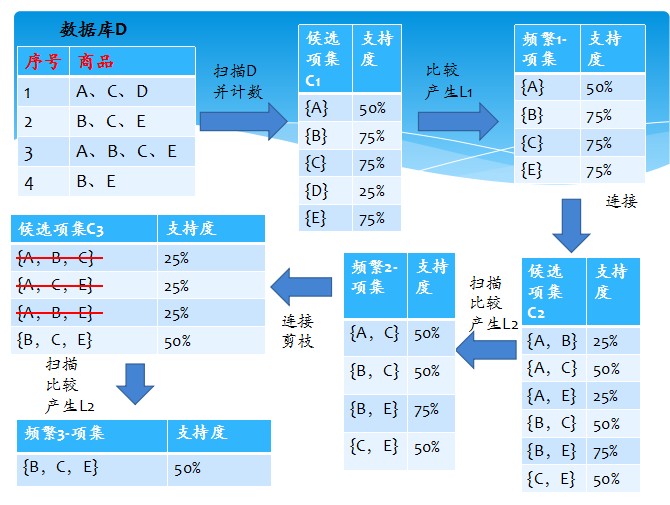
（1）对于每个频繁项集L，产生L的所有非空子集；

（2）对于L的每个非空子集S，如果

                P（L）/P（S）≧min\_conf（置信度）

则输出规则“S L-S”

注：L-S表示在项集L中除去S子集的项集。



#### 总结

这个算法比较简单，缺点是频繁集有多少项就要扫描多少次数据库。

海量数据下，Apriori算法的时空复杂度都不容忽视。

空间复杂度：如果[clip_image010](http://hi.csdn.net/attachment/201106/2/0_130697600694Ea.gif)数量达到[clip_image012](http://hi.csdn.net/attachment/201106/2/0_1306976007fe7O.gif)的量级，那么[clip_image014](http://hi.csdn.net/attachment/201106/2/0_1306976007clpz.gif)中的候选项将达到[clip_image016](http://hi.csdn.net/attachment/201106/2/0_1306976008nDr4.gif)的量级，因为是排列组合公式为阶乘形式。

时间复杂度：每计算一次[clip_image018](http://hi.csdn.net/attachment/201106/2/0_1306976008jtAN.gif)就需要扫描一遍数据库。

### 2、[FP-tree](http://baike.baidu.com/view/3342544.htm" \t "_blank)算法

#### 算法介绍

[**FP-tree**](http://baike.baidu.com/view/3342544.htm)**算法(Frequent Pattern-tree频繁模式树)**使用了一种紧缩的数据结构来存储查找频繁项集所需要的全部信息。

FP- tree算法比Apriori算法快一个数量级，在空间复杂度方面也比Apriori也有数量级级别的优化。但是对于海量数据，FP- tree的时空复杂度仍然很高，可以采用的改进方法包括数据库划分，数据采样等等。

#### 算法步骤

我们举个例子来详细讲解FP-Tree算法的完整实现。

事务数据库如下，一行表示一条购物记录：

牛奶，鸡蛋，面包，薯片

鸡蛋，爆米花，薯片，啤酒

鸡蛋，面包，薯片

牛奶，鸡蛋，面包，爆米花，薯片，啤酒

牛奶，面包，啤酒

鸡蛋，面包，啤酒

牛奶，面包，薯片

牛奶，鸡蛋，面包，黄油，薯片

牛奶，鸡蛋，黄油，薯片

我们的目的是要找出哪些商品总是相伴出现的，比如人们买薯片的时候通常也会买鸡蛋，则[薯片，鸡蛋]就是一条频繁模式（frequent pattern）。

第一步：

扫描事务数据库，每项商品按频数递减排序，并删除频数小于最小支持度MinSup的商品。（第一次扫描数据库）

薯片:7鸡蛋:7面包:7牛奶:6啤酒:4 （这里我们令MinSup=3）

以上结果就是频繁1项集，记为F1。

第二步：

对于每一条购买记录，按照F1中的顺序重新排序。（第二次也是最后一次扫描数据库）

薯片,鸡蛋,面包,牛奶

薯片,鸡蛋,啤酒

薯片,鸡蛋,面包

薯片,鸡蛋,面包,牛奶,啤酒

面包,牛奶,啤酒

鸡蛋,面包,啤酒

薯片,面包,牛奶

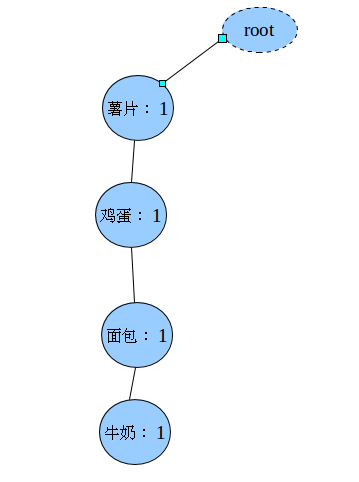
薯片,鸡蛋,面包,牛奶

薯片,鸡蛋,牛奶

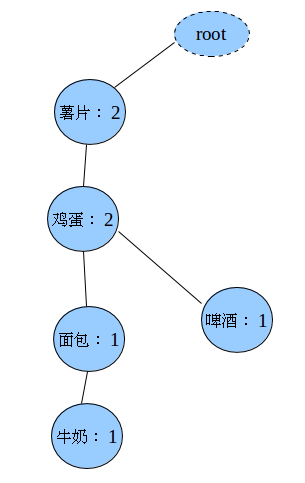
第三步：

把第二步得到的各条记录插入到FP-Tree中。刚开始时后缀模式为空。

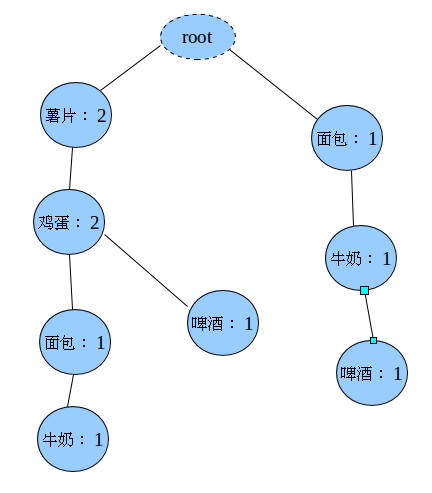
插入第一条（薯片,鸡蛋,面包,牛奶）之后



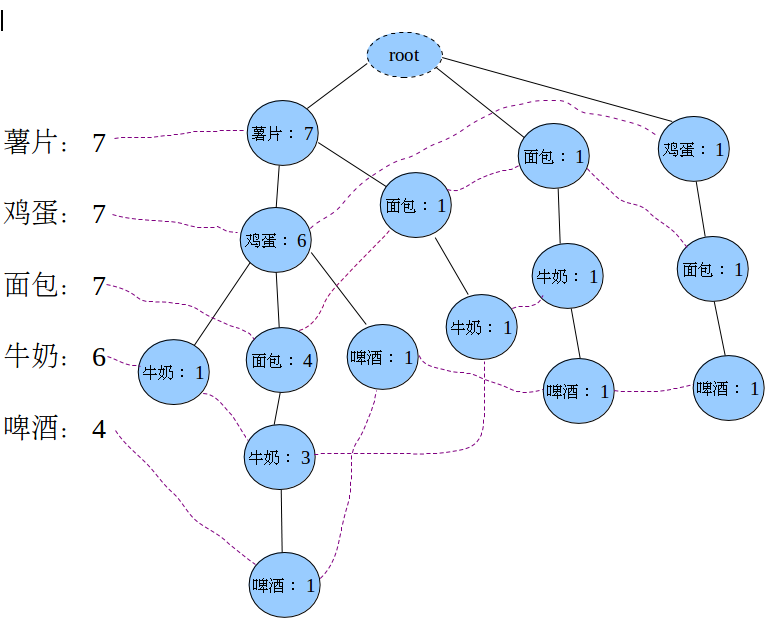
插入第二条记录（薯片,鸡蛋,啤酒）



插入第三条记录（面包,牛奶,啤酒）



最终生成的FP-Tree是



第四步：从FP-Tree中找出频繁项。

遍历表头项中的每一项（我们拿“牛奶：6”为例），对于各项都执行以下（1）到（5）的操作：

（1）从FP-Tree中找到所有的“牛奶”节点，向上遍历它的祖先节点，得到4条路径：

薯片：7，鸡蛋：6，牛奶：1

薯片：7，鸡蛋：6，面包：4，牛奶：3

薯片：7，面包：1，牛奶：1

面包：1，牛奶：1

对于每一条路径上的节点，其count都设置为牛奶的count

薯片：1，鸡蛋：1，牛奶：1

薯片：3，鸡蛋：3，面包：3，牛奶：3

薯片：1，面包：1，牛奶：1

面包：1，牛奶：1

因为每一项末尾都是牛奶，可以把牛奶去掉，得到条件模式基（Conditional Pattern Base,CPB），此时的后缀模式是：（牛奶）。

薯片：1，鸡蛋：1

薯片：3，鸡蛋：3，面包：3

薯片：1，面包：1

面包：1

（2）我们把上面的结果当作原始的事务数据库，返回到第3步，递归迭代运行。每次递归介绍后缀就会变长，直到所有递归结束。

#### 总结

FP-tree减少了数据库的读取次数，但却在机器内存里递归建造类似哈弗曼的模式树。但数据量过大时内存无法容下一棵树，解决方法有集群计算。

### 3、集群计算

Hadoop实现了一个[分布式文件系统](http://baike.baidu.com/view/771589.htm" \t "_blank)（Hadoop Distributed File System），简称HDFS。HDFS有高[容错性](http://baike.baidu.com/view/2700299.htm" \t "_blank)的特点，并且设计用来部署在低廉的（low-cost）硬件上；而且它提供高吞吐量（high throughput）来访问[应用程序](http://baike.baidu.com/view/330120.htm" \t "_blank)的数据，适合那些有着超大数据集（large data set）的应用程序。HDFS放宽了（relax）POSIX的要求，可以以流的形式访问（streaming access）文件系统中的数据。

Hadoop的框架最核心的设计就是：HDFS和MapReduce。**HDFS为海量的数据提供了存储，则MapReduce为海量的数据提供了计算。**

## 三、算法实现

以上为商品关联原理，第三部分是关于算法实现。算法实现步骤：

1. 数据库得到**输入数据**
2. 编程语言对数据进行**程序处理**
3. 返回**输出数据**给数据库

以下为算法测试过程。

### 1、运行环境

编程语言：

**Java**(TM) SE Runtime Environment (build 1.7.0\_15-b03)

Java HotSpot(TM) 64-Bit Server VM (build 23.7-b01, mixed mode)

数据库：

Server version: 5.6.17 **MySQL** Community Server (GPL)

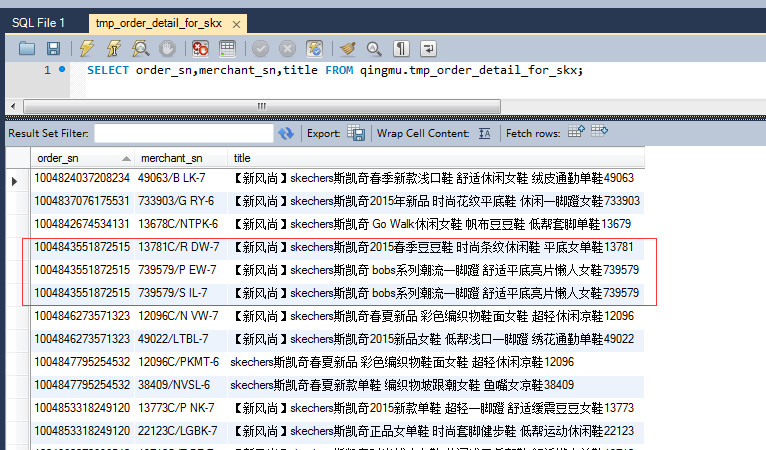
操作系统：64 bit Windows 7旗舰版,内存4G,CPU 2.80GHz

### 2、输入数据

原始数据库表如下，重要的字段是**order\_sn订单编码，merchant\_sn商家编号，order\_status订单状态**

**CREATE TABLE** `tmp\_order\_detail\_for\_skx` (  
 **`order\_sn` varchar**(32) **NOT NULL** COMMENT **'订单编号'**,  
 **`merchant\_sn` varchar**(32) **NOT NULL** COMMENT **'商家编码'**,  
 **`order\_status` int**(11) **NOT NULL** COMMENT **'订单状态'**,  
 **`title` varchar**(512) **DEFAULT NULL** COMMENT **'标题'**,  
 **`price` decimal**(8,2) **DEFAULT NULL** COMMENT **'价格'**,  
 **`amount` int**(11) **DEFAULT NULL** COMMENT **'购买数量'**,  
 **`exsys\_sn` varchar**(32) **DEFAULT NULL** COMMENT **'外部系统编号'**,  
 **`goods\_attri` varchar**(128) **DEFAULT NULL** COMMENT **'商品属性'**,  
 **`package\_info` varchar**(256) **DEFAULT NULL** COMMENT **'套餐信息'**,  
 **`remark` varchar**(512) **DEFAULT NULL** COMMENT **'备注'**,  
 **`import\_date` date DEFAULT NULL** COMMENT **'导入时间YYYY-MM-DD'**,  
 **`data\_status` int**(11) **DEFAULT '0'** COMMENT **'数据状态'**,  
 **`shop\_id` int**(11) **NOT NULL**,  
 **PRIMARY KEY** (**`order\_sn`**,**`merchant\_sn`**,**`order\_status`**)  
) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

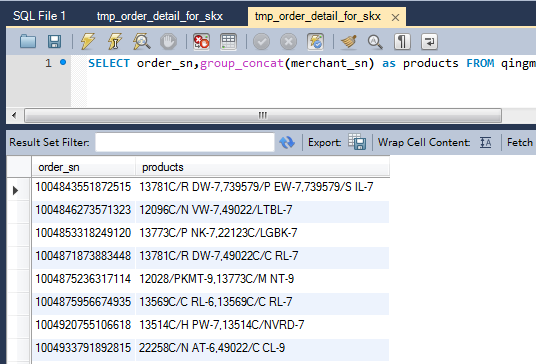
测试数据为**skechers官方旗舰店** 的2015年5月14日-9月14日四个月的历史数据，总共**494405条记录。**如下图红框部分的商品属于同一个订单。



抽取订单状态为1（表示交易成功），订单多于两件商品的所有数据。

**SELECT** order\_sn,group\_concat(merchant\_sn) **as** products **FROM** qingmu.tmp\_order\_detail\_for\_skx **where** order\_status=1 **group by** order\_sn **having** *count*(*\**)>=2

结果如下：



由于是测试数据，且**该店铺编号混乱，**只需**截取/之前**的编码部分，如：

13781C/R DW-7,739579/P EW-7,739579/S IL-7截取后为 **13781C,739579**

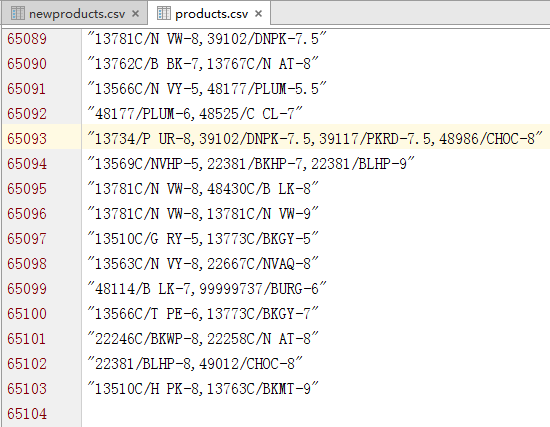
**某些订单还可能因为截取后都是购买相同类型商品，所以数据无效，应该剔除。**

**强烈要求增加类目字段，则数据处理步骤可省略。**

先导出csv格式文件方便去重，数据量大时此种做法不可取。保存为csv文件后，使用python语言进行去重，生成新的csv文件用于后继算法操作。

**SELECT** group\_concat(merchant\_sn) **as** products **FROM** qingmu.tmp\_order\_detail\_for\_skx **where** order\_status=1 **group by** order\_sn **having** *count*(*\**)>=2 limit 0 ,1000000 **into** outfile **'D://products.csv'**fields terminated **by ','** optionally enclosed **by '"'** escaped **by '"'** lines terminated **by '\r\n'**;

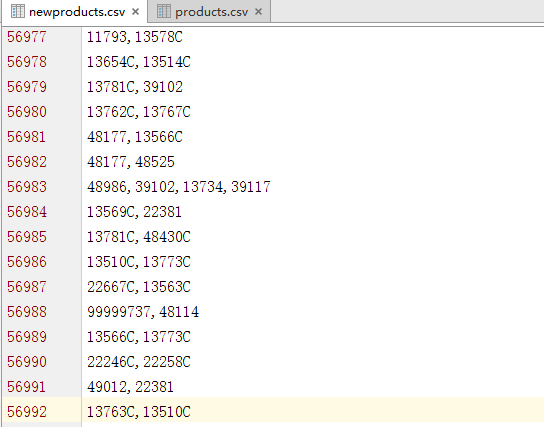
数据库导出文件，商品大于两件的订单数为**65103**



Python去重处理

*# -\*- coding:utf-8 -\*-***import** re  
*# 数据文件*filename = **'products.csv'***# 读取文件*original = open(filename,**'r'**)  
*# 替换双引号*temp = original.read().replace(**'"'**,**''**)  
original.close()  
*# 写入文件*datas=temp.split(**'\n'**)  
*# 正则匹配*rep = re.compile(**r'(.\*)/.\*'**)  
**for** i **in** range(0,len(datas)-1):  
 temp1 = datas[i].split(**','**)  
 **for** j **in** range(0,len(temp1)):  
 **try**:  
 temp1[j]=rep.match(temp1[j]).group(1)  
 *# 出现没有/的商品 剔除掉* **except**:  
 **pass** *# 商品去重* temp1=list(set(temp1))  
 *# 如果只有一件商品设空* **if** len(temp1)==1:  
 datas[i]=**''  
 else**:  
 datas[i]=**','**.join(temp1)  
*# 去重***while '' in** datas:  
 datas.remove(**''**)  
*# 写入新文件*temp=**'\n'**.join(datas)  
newfilename = **'newproducts.csv'**original = open(newfilename,**'w'**)  
original.write(temp)  
original.close()

生成的新文件订单数为**56992**，因为截取之后某些值重复了，剔除掉。



最终生成的文件作为程序运行输入数据。

### 3、程序处理

由于要减少数据库读写次数，我们选择了**FP-tree算法**，鉴于购买两件商品后再购买一件商品，如 {A，B}🡺{C} 也已经计算在 {A}🡺{C}，{B}🡺{C} ，所以我们规定左边购物篮只有一件商品，右边也只有一件，并约束支持度大于0.001，减少计算量。全部商品关联规则类似 {左购物篮商品}🡺{右购物篮商品} 。

Java代码如下（由于后继进行达芙妮数据测试，程序数据源分为两种，一种是从数据库数据转移到csv文件，进行数据预处理再运行，第二种是直接使用数据库数据。

#### TreeNode.java

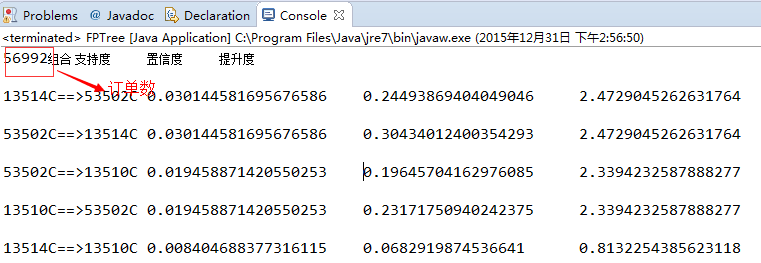
#### Mysql.java

#### FPTree.java

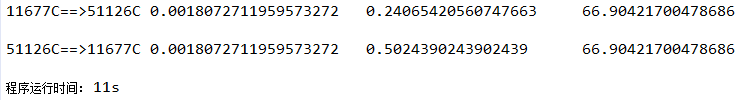
代码略，请见附件。

### 4、输出数据

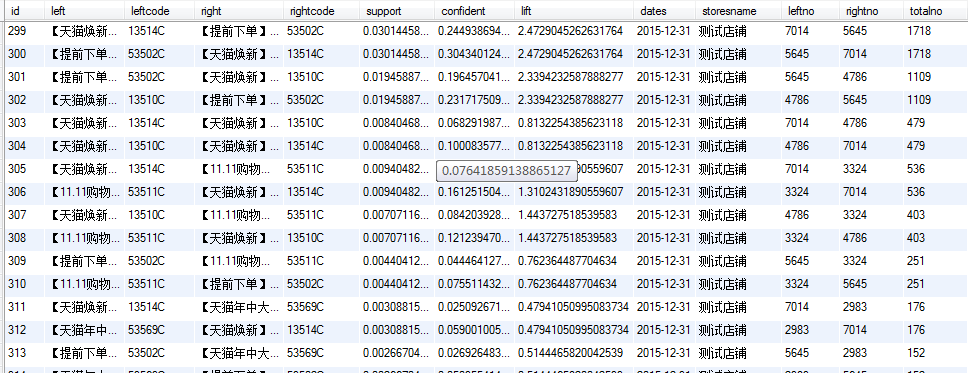
JAVA Eclispe显示台打印出的帮助提示如下，但是我们将生成的数据存在数据库中。



…..



**随着数据量增大，运行时间将会增大，所以我们新建了数据库表保存数据，**方便运行一次，可用多次，即是每个月或每个季度运行一次就足够了。同时，我们**反向**将编号进行数据库查找，查找**最新日期的商品标题**。如下图：



新建的数据库表结构如下：

**CREATE TABLE** `linkresult` (  
 **`id` int**(11) **NOT NULL** AUTO\_INCREMENT,  
 **`left` varchar**(100) **NOT NULL** COMMENT **'左购物篮商品最新标题'**,  
 **`leftcode` varchar**(45) **NOT NULL** COMMENT **'左购物篮商品编码'**,  
 **`right` varchar**(100) **NOT NULL** COMMENT **'右购物篮商品最新标题'**,  
 **`rightcode` varchar**(45) **NOT NULL** COMMENT **'右购物篮商品编码'**,  
 **`support` double NOT NULL** COMMENT **'支持度'**,  
 **`confident` double NOT NULL** COMMENT **'置信度'**,  
 **`lift` double NOT NULL** COMMENT **'提升度'**,  
 **`dates` date DEFAULT NULL** COMMENT **'处理时间'**,  
 **`storesname` varchar**(100) **DEFAULT NULL** COMMENT **'店铺名'**,  
 **`leftno` int**(11) **NOT NULL** COMMENT **'左边商品订单数'**,  
 **`rightno` int**(11) **NOT NULL** COMMENT **'右边商品订单数'**,  
 **`totalno` int**(11) **NOT NULL** COMMENT **'组合商品订单数'**,  
 **PRIMARY KEY** (**`id`**)  
) ENGINE=InnoDB AUTO\_INCREMENT=597 **DEFAULT** CHARSET=utf8 COMMENT=**'关联结果'**;

## 四、结果分析

总计56992个有效订单中

支持度Top1组合如下

**【天猫焕新】skechers斯凯奇舒适一脚蹬 超轻豆豆懒人鞋 透气网布女单鞋**

类目编号：13514C 订单数：7014

**【提前下单】Skechers斯凯奇男女款健步鞋 透气情侣鞋赤足鞋1351**0

类目编号：53502C 订单数:5645

有1718个多商品组合订单同时购买了这两件商品。

**【天猫焕新】skechers斯凯奇舒适一脚蹬 超轻豆豆懒人鞋 透气网布女单鞋**

**【提前下单】Skechers斯凯奇男女款健步鞋 透气情侣鞋赤足鞋13510**

购买了女单鞋类型后再购买情侣鞋类型的置信度高达0.244，可以推荐出售。

|  |  |
| --- | --- |
| **支持度** | 0.030145 |
| **置信度** | 0.244939 |
| **提升度** | 2.472905 |

支持度前八个的组合如下表。

表：商品关联支持度Top8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **左购物篮标题** | 【天猫焕新】skechers斯凯奇舒适一脚蹬 超轻豆豆懒人鞋 透气网布女单鞋13514 | 【提前下单】Skechers斯凯奇男女款健步鞋 透气情侣鞋赤足鞋13510 | 【提前下单】Skechers斯凯奇男女款健步鞋 透气情侣鞋赤足鞋13510 | 【天猫焕新】skechers斯凯奇单鞋 低帮透气懒人健步鞋 超轻舒适豆豆女鞋13510 | 【天猫焕新】skechers斯凯奇秋季新款懒人鞋 绒毛内里豆豆鞋 保暖女鞋13533 | 【99周年庆】skechers斯凯奇新款秋冬男鞋 加绒套脚豆豆鞋53585 | 【天猫年中大促商品】Skechers斯凯奇 天猫正品情侣鞋 2013新款圆头男女款鞋户外13569 | 【天猫焕新】skechers斯凯奇休闲健步鞋 舒适系带豆豆鞋 透气跑步女鞋13569 |
| **左购物篮编码** | 13514C | 53502C | 53502C | 13510C | 13533C | 53585C | 53569C | 13569C |
| **右购物篮标题** | 【提前下单】Skechers斯凯奇男女款健步鞋 透气情侣鞋赤足鞋13510 | 【天猫焕新】skechers斯凯奇舒适一脚蹬 超轻豆豆懒人鞋 透气网布女单鞋13514 | 【天猫焕新】skechers斯凯奇单鞋 低帮透气懒人健步鞋 超轻舒适豆豆女鞋13510 | 【提前下单】Skechers斯凯奇男女款健步鞋 透气情侣鞋赤足鞋13510 | 【99周年庆】skechers斯凯奇新款秋冬男鞋 加绒套脚豆豆鞋53585 | 【天猫焕新】skechers斯凯奇秋季新款懒人鞋 绒毛内里豆豆鞋 保暖女鞋13533 | 【天猫焕新】skechers斯凯奇休闲健步鞋 舒适系带豆豆鞋 透气跑步女鞋13569 | 【天猫年中大促商品】Skechers斯凯奇 天猫正品情侣鞋 2013新款圆头男女款鞋户外13569 |
| **右购物篮编码** | 53502C | 13514C | 13510C | 53502C | 53585C | 13533C | 13569C | 53569C |
| **支持度** | 0.030145 | 0.030145 | 0.019459 | 0.019459 | 0.013212 | 0.013212 | 0.010458 | 0.010458 |
| **置信度** | 0.244939 | 0.30434 | 0.196457 | 0.231718 | 0.271058 | 0.351213 | 0.199799 | 0.25812 |
| **提升度** | 2.472905 | 2.472905 | 2.339423 | 2.339423 | 7.205296 | 7.205296 | 4.931545 | 4.931545 |
| **处理时间** | 2015/12/31 | 2015/12/31 | 2015/12/31 | 2015/12/31 | 2015/12/31 | 2015/12/31 | 2015/12/31 | 2015/12/31 |
| **处理店铺** | 测试店铺 | 测试店铺 | 测试店铺 | 测试店铺 | 测试店铺 | 测试店铺 | 测试店铺 | 测试店铺 |
| **左购物篮订单数** | 7014 | 5645 | 5645 | 4786 | 2778 | 2144 | 2983 | 2309 |
| **右购物篮订单数** | 5645 | 7014 | 4786 | 5645 | 2144 | 2778 | 2309 | 2983 |
| **左右组合订单数** | 1718 | 1718 | 1109 | 1109 | 753 | 753 | 596 | 596 |

待编码表整理后，结合业务人员分析，可以进行商品推荐系统的设计。

## 五、参考

算法参考

<http://blog.csdn.net/sealyao/article/details/6460578>

<http://blog.sina.com.cn/s/blog_4b62a6270100frx9.html>

<http://www.cnblogs.com/zgw21cn/archive/2009/05/31/1492809.html>

<http://www.cnblogs.com/zhangchaoyang/articles/2198946.html>

书籍：

<https://detail.tmall.com/item.htm?id=27478652515>       高性能MySQL（第3版）  
<https://detail.tmall.com/item.htm?id=45458357369>      Python Cookbook（第3版）  
<https://detail.tmall.com/item.htm?id=523082356946>    语言数据分析与挖掘实战  
<https://detail.tmall.com/item.htm?id=21285963476>    数据挖掘与数据化运营实战  
<https://detail.tmall.com/item.htm?id=23618240149>     数据挖掘技术（第3版)

# 达芙妮官方旗舰店 商品关联案例分析

1. 店铺介绍

Daphne/达芙妮—中国快时尚女鞋第一品牌，自1992年以来连续多年获颁赠最畅销国内产品之荣衔。

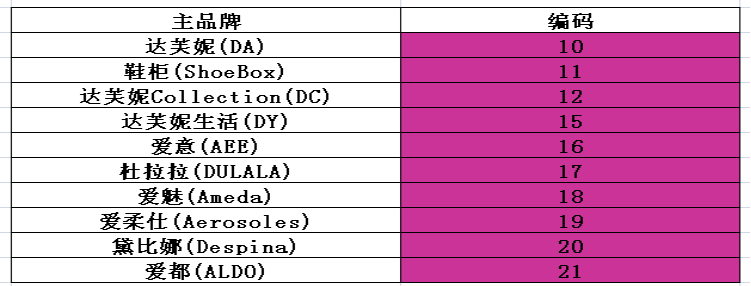
科技2014年全盘接手达芙妮集团旗下的达芙妮官方旗舰店、鞋柜官方旗舰店，从全面梳理货品结构开始着手，进行全新的货品规划与运营节奏、价格体系制定、促销活动策划以及主销款打造，加快上新与运营推广节奏，一举扭转下滑态势，从接手第2个月开始，2家旗舰店均开始呈现出强劲的增势，日销均实现翻番，在2014年双11成功登顶女鞋类目，成为当日销售额最高女鞋店铺。



## 经营商品

达芙妮经营的商品有严格的编码规范，如下图：





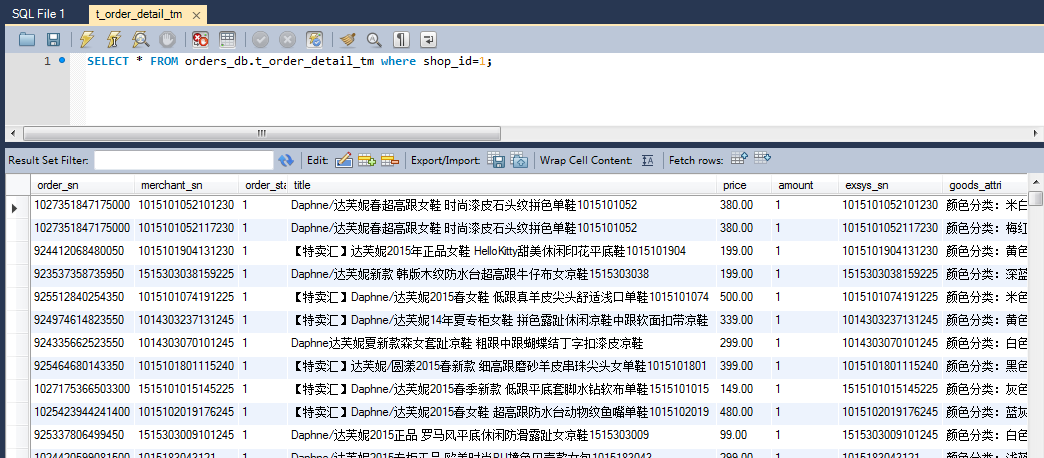






## 数据情况

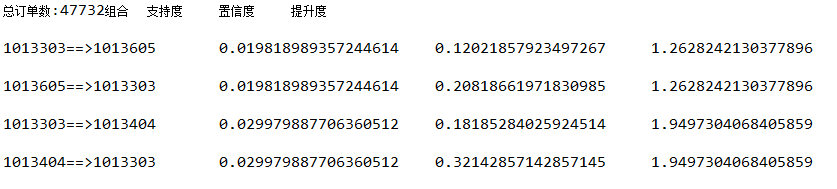
目前数据库中达芙妮订单数据有1643823条记录。数据库数据格式如下：



抽取订单状态为交易成功的多商品订单数据，对这些商品截取前面七位编码：



以下为程序处理控制台打印出来的消息：



可以看出数据预处理后剩下47732个订单数，相比100多万的数据明显减少很多。

对于其他店铺，要根据其编号进行相应数据预处理

## 结果分析

我们选取了支持度为0.0001进行程序处理，出现4次以上的组合都会被处理后记录到数据库，因此处理时间特别长，长达1443秒，约24分钟，如果想节省时间，可将支持度较小一点。

我们选取支持度大于等于0.01的组合进行分析，这意味者这些组合在47732个订单中至少出现了477次，满足分析标准。以下为这些组合：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **左购物篮标题** | Daphne/达芙妮新款厚底女鞋 Hello Kitty甜美条纹坡跟防水台拖鞋 | 【新风尚价】Daphne/达芙妮2014春女鞋 金属装饰羊皮尖头浅口单鞋粗低跟正装鞋 | Daphne/达芙妮欧美休闲女鞋 金属装饰粗跟单靴子 时尚拼接高筒靴 | Daphne/达芙妮头层牛皮女鞋 时尚尖头短毛绒 金属装饰棉靴中筒靴 | Daphne/达芙妮专柜凉鞋 夏季甜美淑女漆皮T结构撞色锥形跟女鞋 | Daphne达芙妮新款女鞋 拼接编织纹坡跟高跟鞋 丁字式扣带鱼嘴凉鞋 | DDaphne达芙妮夏新款女鞋 松糕跟厚底单鞋 撞色鱼嘴凉鞋高跟鞋 | Daphne达芙妮休闲日系女靴 铆钉金属方形扣圆头靴 方跟中跟马丁靴 |
| **左购物篮编码** | 1014303 | 1014101 | 1013605 | 1013607 | 1013303 | 1013103 | 1013303 | 1013605 |
| **右购物篮标题** | 【新风尚价】Daphne/达芙妮2014春女鞋 金属装饰羊皮尖头浅口单鞋粗低跟正装鞋 | Daphne/达芙妮新款厚底女鞋 Hello Kitty甜美条纹坡跟防水台拖鞋 | Daphne/达芙妮头层牛皮女鞋 时尚尖头短毛绒 金属装饰棉靴中筒靴 | Daphne/达芙妮欧美休闲女鞋 金属装饰粗跟单靴子 时尚拼接高筒靴 | Daphne达芙妮新款女鞋 拼接编织纹坡跟高跟鞋 丁字式扣带鱼嘴凉鞋 | Daphne/达芙妮专柜凉鞋 夏季甜美淑女漆皮T结构撞色锥形跟女鞋 | Daphne达芙妮休闲日系女靴 铆钉金属方形扣圆头靴 方跟中跟马丁靴 | DDaphne达芙妮夏新款女鞋 松糕跟厚底单鞋 撞色鱼嘴凉鞋高跟鞋 |
| **右购物篮编码** | 1014101 | 1014303 | 1013607 | 1013605 | 1013103 | 1013303 | 1013605 | 1013303 |
| **支持度** | 0.012968239 | 0.012968239 | 0.015251823 | 0.015251823 | 0.015817481 | 0.015817481 | 0.019818989 | 0.019818989 |
| **置信度** | 0.147521449 | 0.411842981 | 0.160211268 | 0.447174447 | 0.095946118 | 0.655381944 | 0.120218579 | 0.20818662 |
| **提升度** | 4.684959284 | 4.684959284 | 4.697299893 | 4.697299893 | 3.975434105 | 3.975434105 | 1.262824213 | 1.262824213 |
| **左购物篮订单数** | 4196 | 1503 | 4544 | 1628 | 7869 | 1152 | 7869 | 4544 |
| **右购物篮订单数** | 1503 | 4196 | 1628 | 4544 | 1152 | 7869 | 4544 | 7869 |
| **左右组合订单数** | 619 | 619 | 728 | 728 | 755 | 755 | 946 | 946 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **左购物篮标题** | Daphne/达芙妮欧美休闲女鞋 金属装饰粗跟单靴子 时尚拼接高筒靴 | Daphne/达芙妮2014专柜正品纯色女鞋 素色石头纹尖头粗跟单鞋 | DDaphne达芙妮夏新款女鞋 松糕跟厚底单鞋 撞色鱼嘴凉鞋高跟鞋 | Daphne/达芙妮新款厚底女鞋 Hello Kitty甜美条纹坡跟防水台拖鞋 | DDaphne达芙妮夏新款女鞋 松糕跟厚底单鞋 撞色鱼嘴凉鞋高跟鞋 | Daphne/达芙妮深口单鞋 休闲金属链尖头女式鞋 高档粗跟超高跟鞋 | Daphne/达芙妮专柜凉鞋 夏季甜美淑女漆皮T结构撞色锥形跟女鞋 | Daphne/达芙妮夏款女鞋 撞色漆皮糖果色高跟鞋鱼嘴高跟坡跟凉鞋 |
| **左购物篮编码** | 1013605 | 1013404 | 1013303 | 1014303 | 1013303 | 1013404 | 1013303 | 1213203 |
| **右购物篮标题** | Daphne/达芙妮2014专柜正品纯色女鞋 素色石头纹尖头粗跟单鞋 | Daphne/达芙妮欧美休闲女鞋 金属装饰粗跟单靴子 时尚拼接高筒靴 | Daphne/达芙妮新款厚底女鞋 Hello Kitty甜美条纹坡跟防水台拖鞋 | DDaphne达芙妮夏新款女鞋 松糕跟厚底单鞋 撞色鱼嘴凉鞋高跟鞋 | Daphne/达芙妮深口单鞋 休闲金属链尖头女式鞋 高档粗跟超高跟鞋 | DDaphne达芙妮夏新款女鞋 松糕跟厚底单鞋 撞色鱼嘴凉鞋高跟鞋 | Daphne/达芙妮夏款女鞋 撞色漆皮糖果色高跟鞋鱼嘴高跟坡跟凉鞋 | Daphne/达芙妮专柜凉鞋 夏季甜美淑女漆皮T结构撞色锥形跟女鞋 |
| **右购物篮编码** | 1013404 | 1013605 | 1014303 | 1013303 | 1013404 | 1013303 | 1213203 | 1013303 |
| **支持度** | 0.026942093 | 0.026942093 | 0.029686583 | 0.029686583 | 0.029979888 | 0.029979888 | 0.040853096 | 0.040853096 |
| **置信度** | 0.283010563 | 0.28885894 | 0.180073707 | 0.337702574 | 0.18185284 | 0.321428571 | 0.247807854 | 0.702449568 |
| **提升度** | 3.034290254 | 3.034290254 | 2.048445705 | 2.048445705 | 1.949730407 | 1.949730407 | 4.260938209 | 4.260938209 |
| **左购物篮订单数** | 4544 | 4452 | 7869 | 4196 | 7869 | 4452 | 7869 | 2776 |
| **右购物篮订单数** | 4452 | 4544 | 4196 | 7869 | 4452 | 7869 | 2776 | 7869 |
| **左右组合订单数** | 1286 | 1286 | 1417 | 1417 | 1431 | 1431 | 1950 | 1950 |

总共有16个组合，组合商品的标题是根据类目反查数据库，只具备参考性。

支持度最高的两个组合。

1014303--->1014101 1014101---->1014303

|  |  |  |
| --- | --- | --- |
| **左购物篮标题** | Daphne/达芙妮新款厚底女鞋 Hello Kitty甜美条纹坡跟防水台拖鞋 | 【新风尚价】Daphne/达芙妮2014春女鞋 金属装饰羊皮尖头浅口单鞋粗低跟正装鞋 |
| **左购物篮编码** | 1014303 | 1014101 |
| **右购物篮标题** | 【新风尚价】Daphne/达芙妮2014春女鞋 金属装饰羊皮尖头浅口单鞋粗低跟正装鞋 | Daphne/达芙妮新款厚底女鞋 Hello Kitty甜美条纹坡跟防水台拖鞋 |
| **右购物篮编码** | 1014101 | 1014303 |
| **支持度** | 0.012968239 | 0.012968239 |
| **置信度** | 0.147521449 | 0.411842981 |
| **提升度** | 4.684959284 | 4.684959284 |
| **左购物篮订单数** | 4196 | 1503 |
| **右购物篮订单数** | 1503 | 4196 |
| **左右组合订单数** | 619 | 619 |

可以看出两个组合其实商品是一样的，只不过一个是左购物篮，一个是右购物篮。

通过查找编码规则1014101，1014303

**达芙妮(DA)品牌2014年春季女鞋春单类别**

**达芙妮(DA)品牌2014年酷夏凉鞋类别**

组合卖得最火，支持度最高，荣获商品关联中购买人数最多的组合。

**可以想象夏天已经到了，而春天似乎还没走远，春夏换季人们为了舒适，同时购买了两样商品。建议春季春单和酷夏凉鞋一起搭配出售，效果更好。**

置信度最高的两个组合

|  |  |  |
| --- | --- | --- |
| **左购物篮标题** | Daphne达芙妮新款女鞋 拼接编织纹坡跟高跟鞋 丁字式扣带鱼嘴凉鞋 | Daphne/达芙妮夏款女鞋 撞色漆皮糖果色高跟鞋鱼嘴高跟坡跟凉鞋 |
| **左购物篮编码** | 1013103 | 1213203 |
| **右购物篮标题** | Daphne/达芙妮专柜凉鞋 夏季甜美淑女漆皮T结构撞色锥形跟女鞋 | Daphne/达芙妮专柜凉鞋 夏季甜美淑女漆皮T结构撞色锥形跟女鞋 |
| **右购物篮编码** | 1013303 | 1013303 |
| **支持度** | 0.015817481 | 0.040853096 |
| **置信度** | 0.655381944 | 0.702449568 |
| **提升度** | 3.975434105 | 4.260938209 |
| **左购物篮订单数** | 1152 | 2776 |
| **右购物篮订单数** | 7869 | 7869 |
| **左右组合订单数** | 755 | 1950 |

置信度最高说明关联规则十分的强，超级推荐组合出售。

通过查找编码规则1013103-->1013303 , 1213203-->1013303

**组合**

**达芙妮(DA)品牌2013年度春季凉鞋**

**达芙妮(DA)品牌2013年度酷夏凉鞋**

**------------------------------------------------**

**组合**

**达芙妮(DC)品牌2013年度初夏凉鞋**

**达芙妮(DA)品牌2013年度酷夏凉鞋**

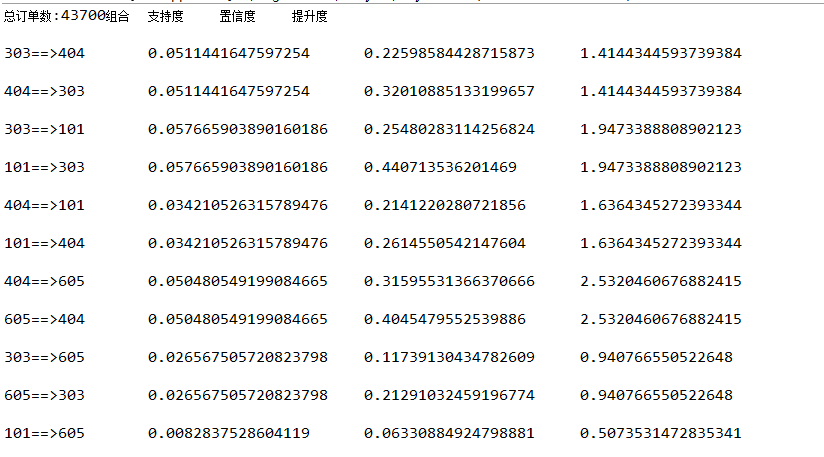
**看出凉鞋绝对是人们出行必备，人们买春季和初夏凉鞋时，必备买酷夏凉鞋，第二个组合置信度高达0.70，说明买了初夏凉鞋的人中，70%的人又同时买了酷夏凉鞋，不怕鞋多，就怕热！所以出售凉鞋时，可以将春天和夏天的凉鞋一起推荐。**

## 五、需求更改后

上述截取粒度过小，故需求更改为截取季节和类别三类编码即可。



程序运行后：



可以看出订单数只减少了一点，从47732变为43700，而由于查找类目名称不再往数据库里反找，时间缩短到53秒，相比24分钟，程序处理速度更快。类目名称请看下文。

结果如下，请放大倍数观看（支持度大于0.01）。

