《数据仓库与数据挖掘技术》项目报告
基于统计方法的数据分布的图形显示

周昱杉

1400012800

1400012800@pku.edu.cn

智能科学与技术系

2017 年 6 月 15 日

目录

摘要

我选择的是 Project1，完成了基本操作。实现了属性数据的分布显示，对数字型数据能够进行离散化，可以自动判断是否需要分箱操作，点击柱状图上属性的某个取值或取值范围可以进一步观察其他属性的取值，并在数据取值不需要分箱的条件下实现归一化；对字符串数据仅根据键进行计数。另外，我虽然没有完成 **time** 和 **date** 的区分，但是能够在 **date** 维度上对另一个数值型属性进行每期的变化率分析。

根据要求，在统计频数时完成了有限制的 **Slice** 和 **Dice** 操作，在时间序列分析时完成了时间维度的 **Drill-down** 和 **Roll-up**。

在完成基本要求的同时，我又尝试了给一个属性对绘制散点图、利用 **FPGrowth** 算法建立 **FPTree** 计算频繁项集。其中，一个属性对是从之前已选好的九个属性中选取两个属性；实现 **FPGrowth** 算法时所用的数据集是所有的数据。

第一章　主要功能与实现方法（For project 1）

1.1　基础功能 1：柱形图显示基于统计方法的数据分布

1.1.1　功能描述

以属性的取值或取值区间作为横坐标，以记录数，即属性在这个取值或取值区间的统计数作为纵坐标，绘制柱形图。

点中属性的某一个属性值/区间，在柱形图中表现为点击某一个柱形块，即可显示其中满足该值/值区间的记录在其他属性中的表示。在图 1.1中点击 marital='married' 代表的块，则得到图 1.2，为在 marital='married' 的记录在其他选中属性下的分布。

1.1.2　实现原理

使用 d3 库，给每个属性加一个 svg 画布，并在画布上作图。画布上需要有的元素为：x-y 坐标轴、属性键值对的块 bar。另外定义了鼠标事件，当鼠标 moveover、mouseout、click 时画布上元素的显示变化。

1.2　基础功能 2：属性数据的离散化

1.2.1　功能描述

这里主要实现的是属性数据的离散化。对单个属性的每一个属性值进行计数，当属性值个数过多时（超过 20 个），进行分箱操作。如图 1.3。由

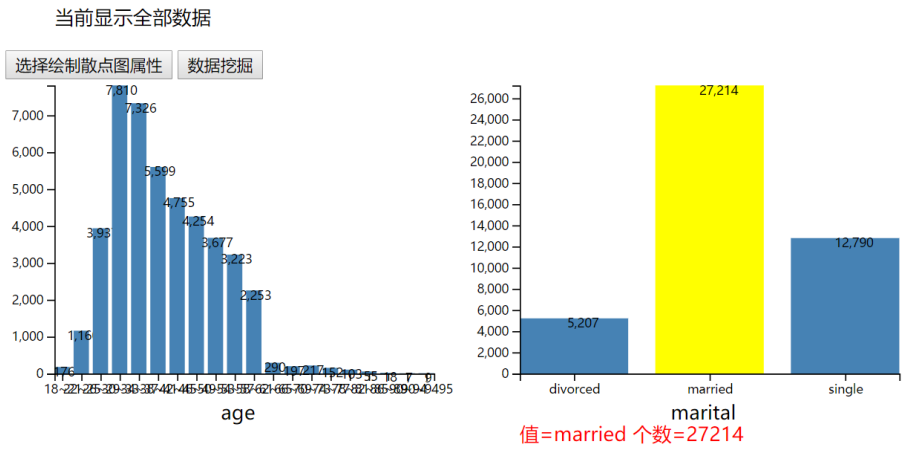

图 1.1: 柱形图显示当前选中的所有属性的属性值分布

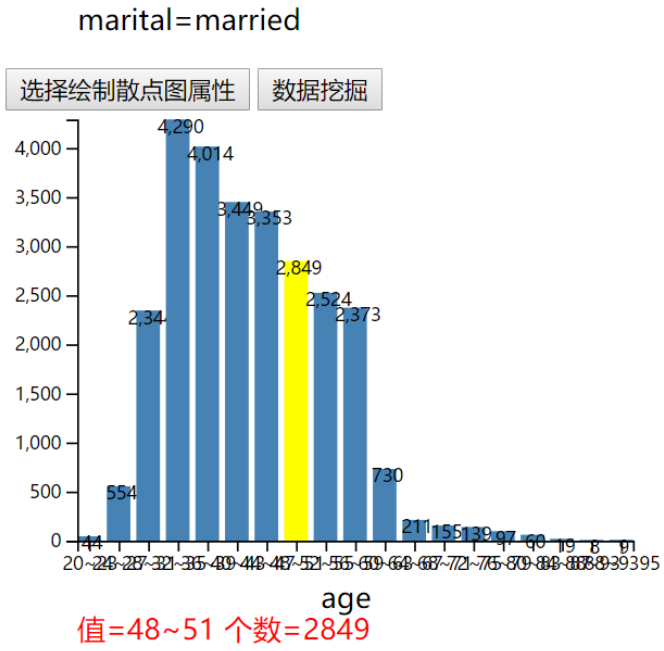

图 1.2: 柱形图显示满足某属性取值的记录在其他属性中的表示

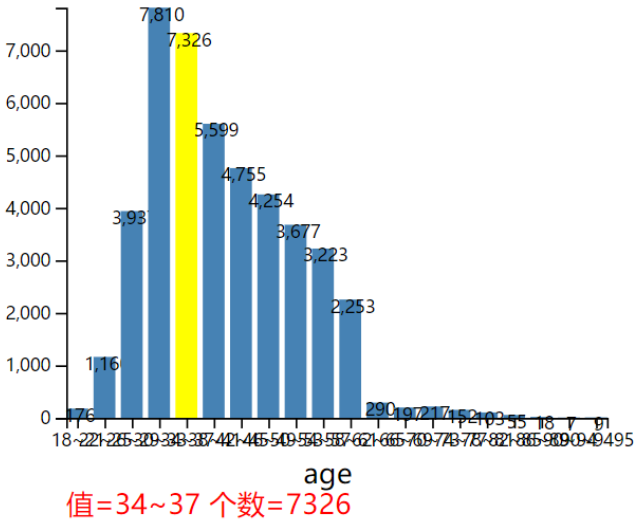

图 1.3: 项目数据集在 age 属性上的计数统计

于数据的表示为取值区间，相邻属性区间空隙较小，故当鼠标移到对应的柱状图的块上时，在柱状图的左下角红字放大显示当前键值对。

1.2.2　实现原理

在绘图的过程中，先确定属性值的个数是否超过了 20。若超过则进行分箱操作，一共分成 20 个箱。同时坐标也有相应的变化，从原来的单取值（数字类型）变为取值区间（字符串）。所以在实际画图中，对数字类型的数据和字符串类型的数据会分别进行处理。

1.3　基础功能 3：属性数据的归一化

1.3.1　功能描述

为了能够清晰表示，我设定当属性值在分箱操作仍旧存在时不能归一化，尽管属性类型为数字，只有当分箱分到最后一步，即在柱状图上的横轴不是取值区间而单个取值时，才能进行归一化。同时，之前选择的属性的取值在左上角显示。如点击图 1.4中的右下角“归一化”，然后就得到图 1.5。图 1.5对当前 duration 值进行了归一化之后的柱状图。此时若将

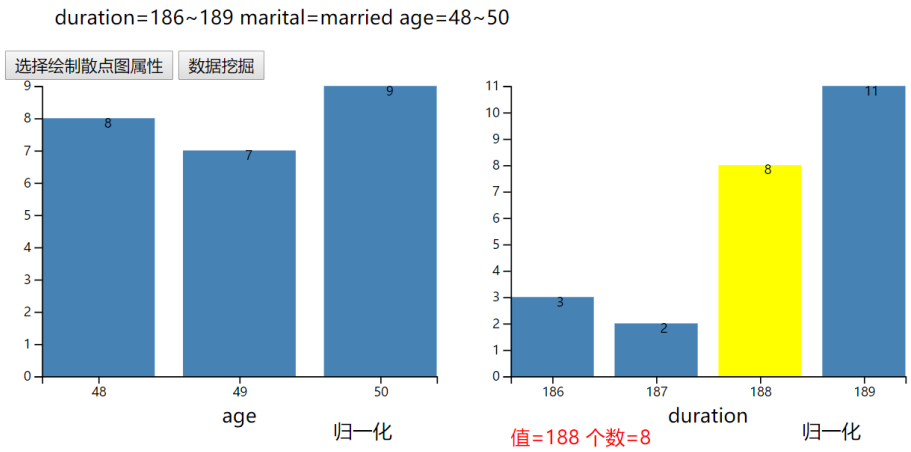

图 1.4: 项目数据集在 duration 属性上的计数统计非归一化

鼠标移到对应的属性值块上，可以发现在左下角显示的键值对中的键仍为原数据，也即，仅在柱状图上显示归一化后的键，本质不变，仍为原数据。图 1.5也可以点击右下角的“反归一化”然后转为图 1.4

1.3.2　实现原理

首先需要确定当前要归一化的数据为数字类型还是字符串类型，若在数据库中类型为数字类型但有经过分箱操作使得坐标为取值区间，也视作字符串类型。

1.4　基础功能 4：数据存放于数据库（MySQL）

1.4.1　功能描述

可以点击主页上的选择文件按钮，选择本地数据集上传到 MySQL 中。数据集要求为，文件名中无特殊字符（如空格等），数据格式为，第一行为属性名，接下来几行为属性取值。

1.4.2　实现原理

主要使用的函数为 fgetcsv。首先先将传进的文件放到一个临时文件夹中，读取文件的第一行得到属性名，确定属性的类型来建立表，选择不多

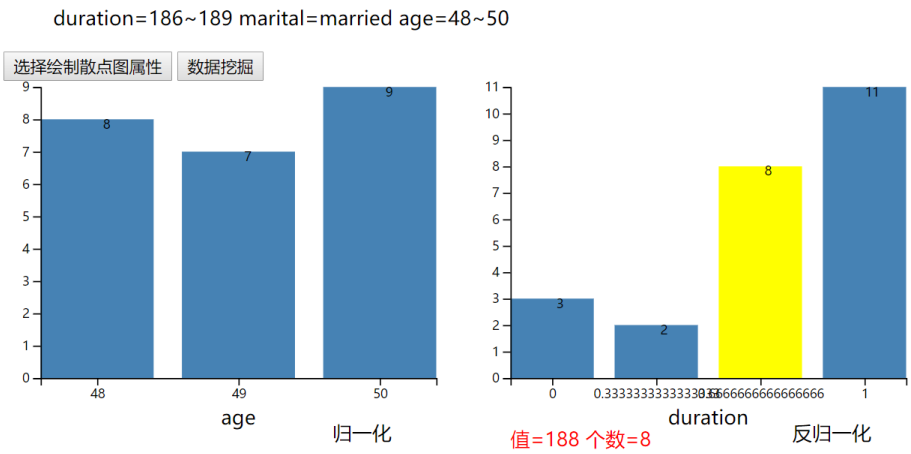

图 1.5: 项目数据集在 duration 属性上的计数统计归一化

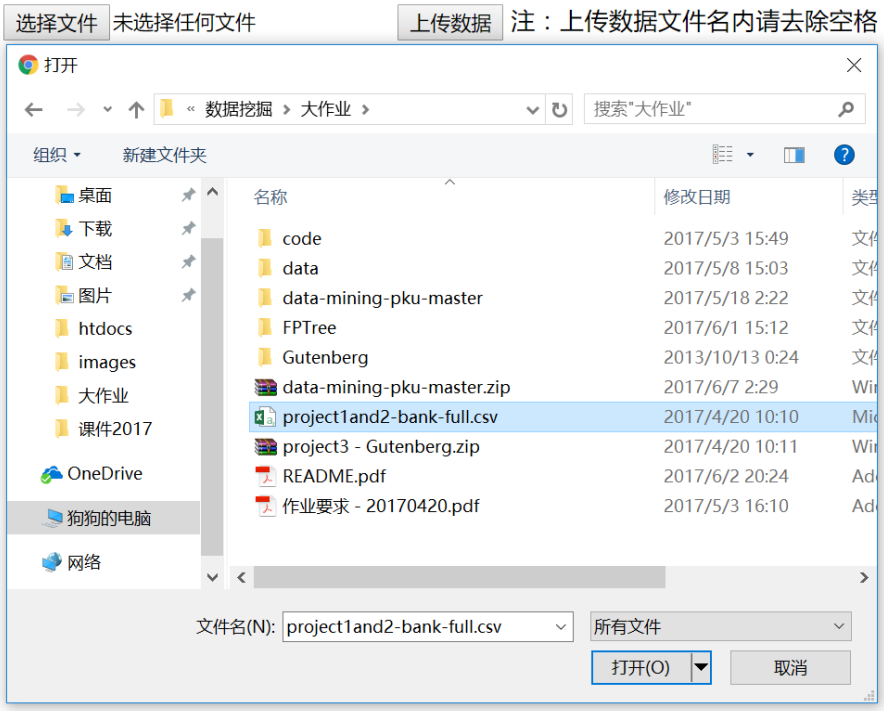

图 1.6: 上传数据

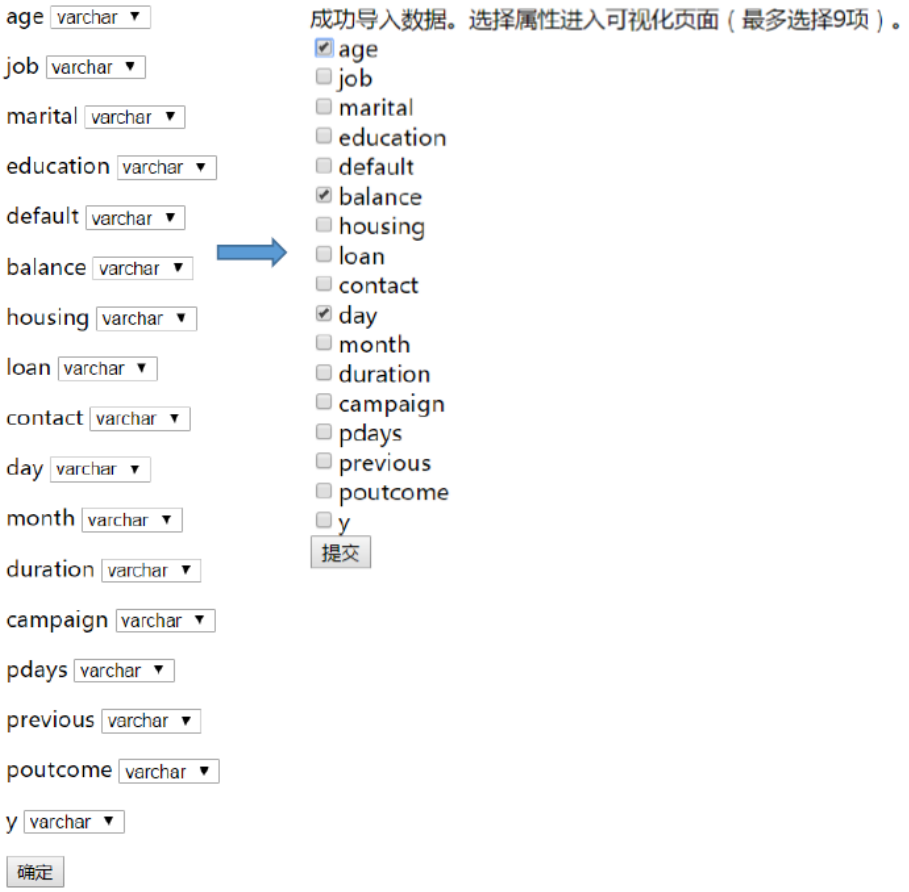

图 1.7: 属性类型确定及选择

于九个属性来绘制柱状图。

1.5　附加功能 5：散点图显示属性对的值分布

1.5.1　功能描述

如图 1.8所示，图为 age 和 balance 属性的散点图，大致反映两个属性之间的分布。此散点图不限制属性类型一定为数值型。散点图的绘制需要用户在中转界面自行选择自上一步遗留下尚未确定具体值（而非值区间）的任意两个属性，属性可相同，虽然没有很大意义。

1.5.2　实现原理

使用 d3 库。

1.6　附加功能 6：在所有数据上运用 FPGrowth 算法

1.6.1　功能描述

在设置了阈值之后，在所有数据（而非之前选择好的不多于九个属性）进行 FPGrowth 算法。如图 1.9所示，点击 FPGrowth 按钮得到得到频繁项集如图 1.10所示，点击 FPTree 得到频繁模式树图 1.11。频繁模式树的每个结点的模式为，属性值 -计数值。点击频繁模式树 Root 结点的子结点 0、no 可进行折叠，得到新的频繁模式树图 1.12。

1.6.2　实现原理

使用 FPGrowth 算法。见下方算法特点与描述。

1.7　加分项功能 7：时间序列分析

1.7.1　功能描述

同散点图类似，在中转界面需要用户自行选择自上一步遗留下尚未确定具体值（而非值区间）的两个属性，其中一个为 date 型属性，一个为数

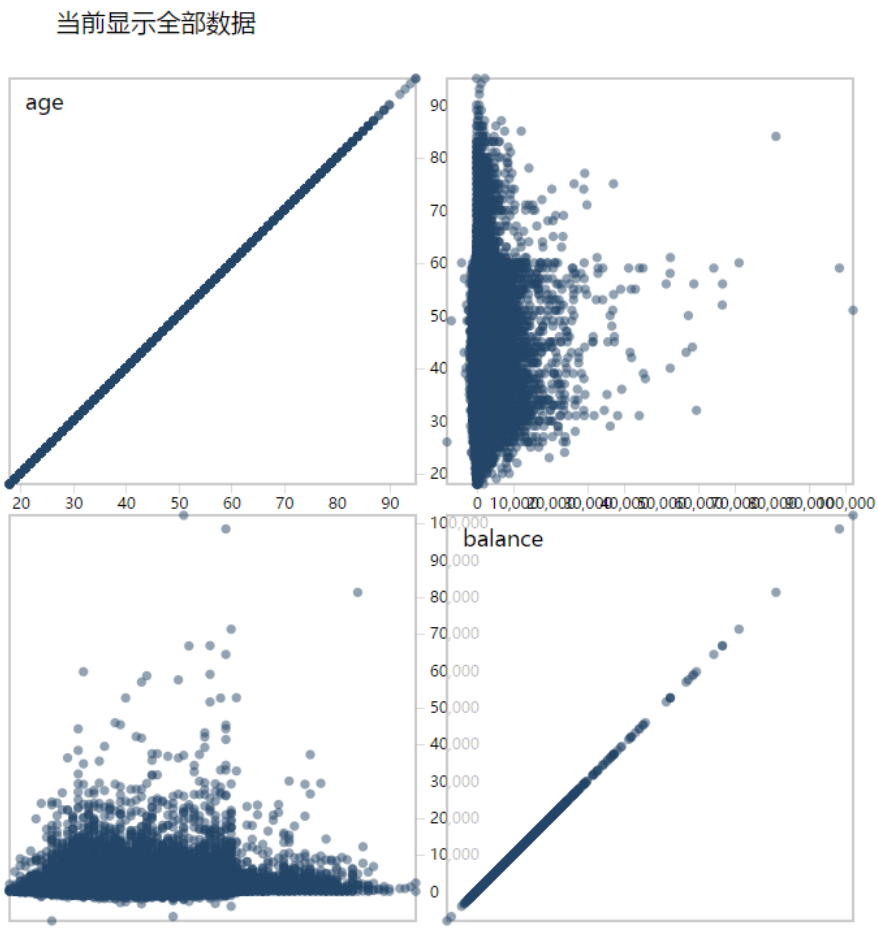

图 1.8: 属性类型确定及选择

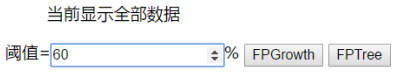

图 1.9: 阈值选择

有频繁项集为：

- married
- married,no
- cellular
- cellular,no
- yes
- yes,no
- -1
- unknown,-1
- unknown,-1,no
- 0,-1
- 0,-1,no
- 0,-1,unknown,no
- 0,-1,unknown
- -1,no
- unknown
- unknown,0
- unknown,no,0
- unknown,no
- 0
- 0,no
- no

图 1.10: 频繁项集

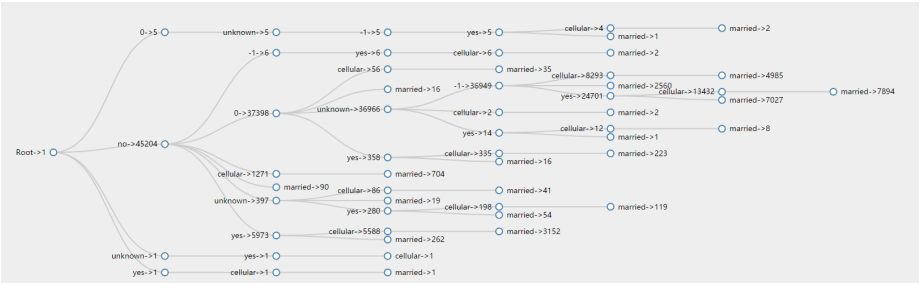

图 1.11: 初始频繁模式树

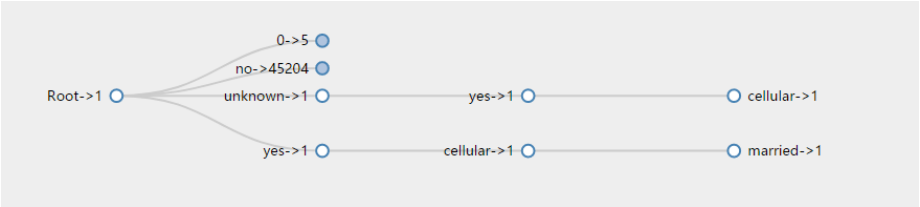

图 1.12: 点击 Root 子结点 0、no 后的频繁模式树

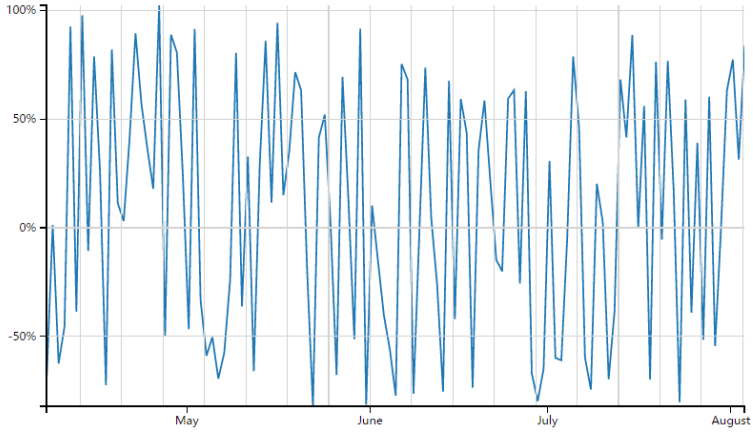

图 1.13: 时间序列分析

值型属性，二者缺一不可，否则页面会提示无法进行绘制。对数值型属性的数据进行这样的操作：若不同的元组有相同 date 型属性的值，那么对数值型属性在此 date 值上做平均。另外，如图 1.13所示具体绘制在图上时为每期平均值的变化率。

1.7.2　实现原理

使用 d3 库。（参考的库为上传代码中的 lib 文件夹）

第二章　算法特点与描述（For project 1）

2.1　算法特点

在本次 project 中，我使用的是 FP-Growth 算法。

在关联分析中，频繁项集的挖掘最常用到的就是 Apriori 算法。Apriori 算法是一种先产生候选项集再检验是否频繁的“产生 -测试”的方法。这种方法有种弊端：当数据集很大的时候，需要不断扫描数据集造成运行效率很低。

而 FP-Growth 算法就很好地解决了这个问题。它的思路是把数据集中的事务映射到一棵 FP-Tree 上面，再根据这棵树找出频繁项集。FP-Tree 的构建过程只需要扫描两次数据集。

FP-Tree 算法构建的频繁项集模式树格式为：

1. 根结点：标为“null”；子树结点：以其父结点项集为前缀的项集；频繁项集头部表。
2. 结点组成：在子树中的每个结点 v 定义以下域：名称：项取值，计数：项计数，连接：结点连接。
3. 频繁项集头部表：在频繁项集头部表中的每个记录 e 定义以下域：名称：项取值，连接头部：指向带有项名称的 FPTree 树的第一个结点的指针。

2.2　算法描述

Algorithm 1 FPTree 构建算法

1: 扫描一遍数据库，得到每个项的支持度 F，根据支持度递减的顺序给项排序得到 Flist，Flist 中的每个项满足最小支持度。
2: 创建一棵频繁模式树 T，建立其根记为“null”。
3: **for** 在数据库中的每个交易 **do**
4: 在交易中挑选出现在 Flist 的项，并根据它在 Flist 中的排名排序。令交易中排过序的项序列记为 [p | P]，p 为第一个元素，P 为剩下的序列
5: **repeat**
6: 将 [p | P] 加入树 T：如果 T 有一个孩子 N，将孩子 N 的计数加一；否则创建一个新节点 N，初始化计数为 1，结点连接初始化为它的父结点连接到 T，通过连接结构连接到同名结点。
7: **until** P 为空
8: **end for**

Algorithm 2 FPGrowth 算法

if 树 T 包含单前缀路径 **then**
  令 P 为树的单前缀路径
  令 Q 为用空根替换顶端分叉结点的多路径部分
  for 对于路径 P 中的每个结点的每个组合 β **do**
    生成模式 $\beta \bigcup v$，v 为 β 中支持度为最小支持度的结点
    令频繁模式集 P 为上述产生的模式集
  end for
else
  令 Q 为树
end if
for 对 Q 中的每个项 item **do**
  生成模式 $\beta \bigcup v$，v 为 β 中支持度为 item 支持度的结点
  构建 β 条件模式基和条件频繁模式树 β_T
  if 树 $\beta_T \neq \emptyset$ **then**
    调用 FPGrowth(β_T,β)
  end if
  令频繁项集 Q 为上述生成的模式
end for
返回频繁项集 set(P)$\bigcup$ 频繁项集 set(Q)$\bigcup$(频繁项集 set(P)$\times$ 频繁项集 set(Q))

第三章　实验

3.1　程序运行环境和操作说明

1. 下载安装 XAMPP，为本级服务器、PHP、MySQL 等整合。
2. 使用的 python 版本为 2.7。由于之前使用的是 3.5，所以将 python2.7 下载之后，可将 python.exe 改名为 python2.exe 。
3. 需要下载 python 包 NumPySciPy、sklearn、matplotlib。
4. 将压缩包内的 htdoc 文件夹替换 xampp 文件夹内的同名 htdocs 文件夹，同名 htdocs 文件夹内的文件夹请保留。htdocs 整个文件夹需要有 apache 的写权限。（或者在`https://github.com/KatrinJo/DataMining`上下载代码并解压缩至 htdocs 文件夹内。）
5. XAMPP 的使用
   (1) 点击 Module 下的 Apache、MySQL 右方 Actions 下的 Start 按钮。
   (2) 点击 Apache 对应的 Admin，或者在浏览器中直接输入 localhost 进入主页。
   (3) 点击 MySQL 对应的 Admin，或者在浏览器中直接输入 localhost/phpmyadmin 进入数据库管理页面。保持原来的数据库，另外新建一个空数据库叫 csv_db。
6. 网页的使用
   (1) 上传数据：上传数据文件类型为 csv（逗号分隔），文件名内无特殊字符（例：空格）。数据格式为第一行为属性名，接下来数行是属性值。属性值不支持中文字符。

(2) 属性类型选择：对第一行属性名进行类型选择。

(3) 属性选择：选择不多于九个属性进行离散化，绘制柱状图。

(4) 单属性（离散化）柱状图：点击柱状图上的块，可以查看在这个属性值（属性区间）内的其余属性值分布。

(5) 双属性散点图：选择两个属性进行散点图绘制。

(6) 数据挖掘：对于所有数据（不仅限于上述九个属性）

i. FPTree：绘制 FPTree。

ii. FPGrowth：展示满足阈值要求的频繁项。

网络资料

- **D3 Data-Driven Documents:** `https://d3js.org/`
- **FP-Growth_Algorithm:** `https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_FP-Growth_Algorithm`
- **My implementation** `https://github.com/KatrinJo/DataMining`