关于消费数据的挖掘和模拟

2021 数学建模 19 级少转大数据小组*

2021 年 6 月 6 日

摘要

本文主要研究了消费者的购物数据特征。我们首先介绍了基于树形结构的两种关联规则挖掘算法 apriori 和 FP-Growth [1]，并将它们运用在 kaggle 提供的 groceries_dataset 数据集上，对该数据集中的用户消费行为进行分析并对两种算法作出比较；然后建立了基于 Markov 过程和随机游走 [2] 的用户购物模型，利用该模型和不同参数分布模拟生成了不同购物数据集，并对其性质进行探究，以判断不同模型与算法生成数据的可信度；最后我们结合生活经验和建模结果为商家提出几点建议，以更好地满足消费者需求或使自己获得更大利益。

关键词：消费　购物　关联规则　Markov 过程　随机游走　模拟数据集

1 问题的提出

DVD 租赁问题[1]是一个经典的数学建模问题，因时代发展，其研究的问题在当下已与时代不符。本次作业基于 DVD 租赁问题，对该问题进行修改，使得其更贴近时代更具有有实际意义。通过本次研究，可以从数据中分析出用户消费特征，从而能给商家销售策略提出具有实际意义的建议。

随着我国经济飞速发展，消费者消费水平不断提高，对消费者购物数据的研究显得越发重要，而挖掘出消费数据的信息对于商家很有帮助。对于该项数据挖掘任务，我们主要有如下两个问题：

(i) 不同商品之间往往存在着关联性，如何通过已有数据发现这种关联性，从而施行更优的销售策略？

(ii) 消费者的消费行为是否有一定特征？能否模拟生成消费数据，并对其合理性做出评价？

通过对这两个问题的研究，我们可以找到一些数据特征，进而对商家提出几点建议。

2 问题分析

问题 (i) 该问题实际上属于早期推荐系统研究中的关联规则挖掘 (Association Rule Mining) 问题，其主要思想是给出事务的集合, 能够发现一些规则——当事务中某些子项出现时，预测其他子项也出现。利用关联规则，商家可以进行合适的捆绑销售，提高商品销量。常用的关联规则算法有 Apriori 和 FP-Growth 两种，它们都基于树形结构。下文中我们将对这两种算法的原理和实验结果进行比较，并分析出和关联规则相关的指标间的联系。

*86 金锦秋 (PB19000167)、87 晏瑞然 (PB19000196)、88 刘子瑞 (PB19000233)

[1]经典 DVD 租赁问题主要考虑的问题如下：有一在线 DVD 租赁平台，顾客缴纳一定数量的月费成为会员，订购 DVD 租赁服务。会员可以在线提交订单租赁自己喜欢的 DVD，网站就会通过快递的方式尽可能满足要求，这些 DVD 是基于其偏爱程度排序。网站会根据手头现有的 DVD 数量和会员的订单进行分发。每个会员租赁次数有一定限制。基于这些假设进行商家购买 DVD 策略的分析。

问题 (ii) 我们假设消费者的一次购物是一次随机游走的过程。具体来说，我们把一次购物分解成很多步，每一步只能选择将一件商品加入购物篮或选择结账，从而结束购物。为了体现出上一问题中商品之间的关联性，我们认为选完某一件商品后，选择下一件商品的概率是服从某个分布的，而不同的分布会得出不同的数据，它们的“逼真度”也不同。具体假定在下文中会详细提出，而我们的主要思想就是把购买过程抽象成一个带吸收壁的 Markov 链。关于评价合理性的问题，我们可以统计生成数据集特征，并将其用在问题 (i) 中的算法中，将相关结果与 groceries dataset 的结果做比较。

3 建模的假设

3.1 关联规则挖掘

- 假设关联规则的存在性成立，即确实有商品与商品之间的联系。
- 设置支持度阈值为 0.001，支持度定义见**数学模型建立**。
- 假设数据量足够的大能满足关联规则挖掘的需求。

3.2 消费者购买行为的模拟

- 消费者在购物时会先选择热门的商品，然后再选择相比之下冷门的物品或者不选。
- 一件商品在单人次的购物中只会被购买一次，即不会出现被重复购买的情况。
- 消费者选择某一件商品的概率（或商品的“热度”）服从某个分布（这里假定商品种类足够多）。
- 基于上面三点假设，购买过程的 Markov 链对应一个上三角矩阵，0 元素代表一次购买结束。

4 符号说明

表 1: **符号说明**

符号	说明	单位
X	项集	1
$\sigma(X)$	项集出现的事务次数	1
$s(X \to Y)$	支持度	1
$c(X \to Y)$	置信度	1
F_k	频繁项集	1
P_{ij}	概率矩阵的元素	1
c_i	购买第 i 个商品不再购买的概率	1
d_i	概率矩阵中的行归一化常数	1

5 数学模型建立

5.1 关联规则挖掘

5.1.1 关联规则相关定义

关联规则可以描述成项集 → 项集。项集 X 出现的事务次数（亦称为 support count）定义为：

$$\sigma(X) = |t_i|\, X \subseteq t_i, t_i \in T\,|$$

其中，t_i 表示某个事务（TID），T 表示事务的集合。关联规则 $X \longrightarrow Y$ 的支持度（support）：

$$s(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{|T|}$$

支持度刻画了项集 $X \bigcup Y$ 的出现频次。置信度（confidence）定义如下：

$$c(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

置信度可理解为条件概率 $p(Y|X)$，度量在已知事务中包含了 X 时包含 Y 的概率。一个好的关联规则，其支持度与置信度均应大于设定的阈值。那么，关联分析问题即等价于：对给定的支持度阈值 min_sup、置信度阈值 min_conf，找出所有的满足下列条件的关联规则：

$$s \geq min\_sup \qquad c \geq min\_conf$$

支持度大于阈值的项集称为频繁项集（frequent itemset）。因此，关联规则分析可分为下列两个步骤：

1) 生成频繁项集 $F = X \cup Y$;

2) 在频繁项集 F 中，找出所有置信度大于最小置信度的关联规则 $X \to Y$ 。

若（对于所有事务集合）项的个数为 d，则所有关联规则的数量为

$$\begin{aligned}
&\sum_{i}^{d} C_d^i \sum_{j}^{d-i} C_{d-i}^j \\
=&\sum_{i}^{d} C_d^i \left(2^{d-i} - 1\right) \\
=&\sum_{i}^{d} C_d^i * 2^{d-i} - 2^d + 1 \\
=&\left(3^d - 2^d\right) - 2^d + 1 \\
=&3^d - 2^{d+1} + 1
\end{aligned}$$

如果采用暴力方法，穷举所有的关联规则，找出符合要求的规则，其时间复杂度将达到指数级。因此，我们需要找出复杂度更低的算法用于关联分析。

5.1.2 Apriori 算法

由定义可以得到如下定理：

定理 1: 如果一个项集是频繁的，那么其所有的子集也一定是频繁的。

由其逆否命题，有：

定理 2: 如果一个项集是非频繁的，那么其所有的超集也一定是非频繁的。

基于此，Apriori 算法从单元素项集开始，通过组合满足最小支持度的项集来形成更大的集合。首先，通过一下两种策略计算大小为 k 的频繁项集 F_k

1) $F_k = F_{k-1} \times F_1$：将所有 F_k 与 F_1 组合;

2) $F_k = F_{k-1} \times F_{k-1}$: 选择前 k−2 项均相同的第 k-1 项进行合并，生成 F_k 。

得到频繁项集后，就需要生成关联规则。

关联规则是由频繁项集生成的，即对于 F_k ，找出项集 h_m ，使得规则 $F_k - h_m \rightarrow h_m$ 的置信度大于置信度阈值。同样地，根据置信度定义得到如下定理：

定理 3: 如果规则 $X \rightarrow Y-X$ 不满足置信度阈值，则对于 X 的子集 X ，规则 $X \rightarrow Y-X$ 也不满足置信度阈值。

通过定理三就可以对频繁项集规则树进行剪枝: 对所有置信度小于置信度阈值的节点, 不再生成新的频繁项集。最后就可以得到所有的满足置信度条件的关联规则: 对所有规则树上的点 X 有 $U - X \rightarrow X$ 其中 U 为全集。

5.1.3 FP-growth 算法

FP-growth 模型利用 FP-tree 这种数据结构大幅降低了复杂度。

定义 1:(FP-tree) 将事务数据表中的各个事务数据项按照支持度排序后，把每个事务中的数据项按降序依次插入到一棵以 NULL 为根结点的树中，同时在每个结点处记录该结点出现的支持度。得到的树称为 FP-tree.

具体构造方法：

首先，遍历一次数据集，统计每个元素出现的次数，然后把出现次数较小的滤掉，然后对每个样本按照元素出现次数重排序。接着，从根节点开始，将过滤并排序后的样本一个个加入树中，若 FP 树不存在现有元素则添加分支，若存在则增加相应的值。具体实现可以用递归的方式完成。最后，为了能方便地访问 FP 树种每一个不同的元素，需要为每种元素（的链表）设置一个头（header），这个 header 除了指向指定元素的第一个结点外，还可以保存该元素在数据集中的总出现次数。

定义 2:(条件模式基) 以所查找元素为结尾的所有前缀路径的集合称为条件模式基。

我们要做的就是从 header 列表开始，针对每一个频繁项，都查找其对应的条件模式基。同时，每一个路径要与起始元素的计数值关联。得到条件模式基后，对每一个频繁项提取了条件模式基，现在就用它作为输入数据，即把每一个前缀路径当成一个样本，通过上述相同的方法构造一棵 FP 树，即条件 FP 树。因创建 FP-tree 的过程中会将所有不满足支持度的节点去除，故最终找到的树的每条分支就是我们所需的频繁项集。最后，对这个条件 FP 树，递归地挖掘。最终就能能获得所有满足最小支持度的频繁项，即我们所需要的频繁项集。

相比 Apriori 算法优点：FP-growth 算法提供了一种相对更快的发现频繁项集的方法。它只遍历 1 次数据集，即可将整个数据集构造成一棵 FP 树，之后从 FP 树中发现频繁项集。提取出频繁项集之后，就可以像 Apriori 算法中的方法一样得到关联规则。

5.2 消费者购买行为的模拟

我们将物品按照热度也就是选取率从高到低进行排序作为状态。假设有 100 个商品分别设其编号为 101 到 200，不购买作为状态 0，购买第 $(100+i)$ 个商品对应状态 i，构造 Markov 链，其概率矩阵为 (P_{ij})。

在问题分析部分中，我们提到过消费者选择某一件商品的概率（或商品的“热度”）服从某个分布。在这里我们以指数分布形式为例，其它分布形式的分析类似。假定购买第一个物品的概率为指数分布密度函数的形式，即

$$P_{0i} = d_0 \lambda e^{-i\lambda}$$

其中 λ 为指数分布的参数。实际上我们的状态并非无穷多，所以按之前的假设最后概率和不为 1，从而设定 d_0 为归一化常数，d_0 满足关系式

$$d_0 = \frac{1}{\sum_{n=0}^{99} \lambda e^{-n\lambda}}.$$

然后对于每一个已经选择过的商品，我们令再选择其他各物品的概率和初始概率成正比，即

$$P_{ij} = kP_{0j} = d_i \lambda e^{-i\lambda}$$

其中 d_i 为归一化常数，满足

$$d_i = \frac{1 - c_i}{\sum_{n=i}^{99} \lambda e^{-n\lambda}}$$

c_i 为每次购买一个物品后不在选择下一个物品的概率，即 $P_{i0} = c_i$. 不难得到 Markov 链的概率矩阵如下所示：

$$\begin{pmatrix}
0 & d_0\lambda & d_0\lambda e^{-\lambda} & d_0\lambda e^{-2\lambda} & d_0\lambda e^{-3\lambda} & \cdots & d_0\lambda e^{-99\lambda} \\
c_1 & 0 & d_1\lambda e^{-\lambda} & d_1\lambda e^{-2\lambda} & d_1\lambda e^{-3\lambda} & \cdots & d_1\lambda e^{-99\lambda} \\
c_2 & 0 & 0 & d_2\lambda e^{-2\lambda} & d_2\lambda e^{-3\lambda} & \cdots & d_2\lambda e^{-99\lambda} \\
c_3 & 0 & 0 & 0 & d_3\lambda e^{-3\lambda} & \cdots & d_3\lambda e^{-99\lambda} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
c_{99} & 0 & 0 & 0 & 0 & \cdots & d_{99}\lambda e^{-99\lambda} \\
c_{100} & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{1}$$

其中 d_i 是归一化常数，c_i 为购买第 i 个商品以后不再购买商品的概率。可以看出除去状态 0，该 Markov 链的概率矩阵为上三角矩阵。

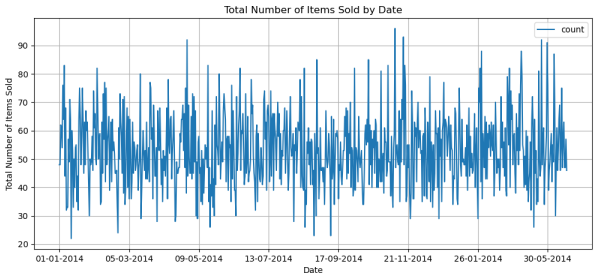
显然如果只有这样的约束，得到的数据一定是完全随机的，这并不是我们想要的！在购买过程中有一些隐性的因素会导致人在选择某一物品后再选择另一个物品的概率会远高出随机选择的概率，比如著名的啤酒与尿布，日常生活中购买乒乓球和乒乓球拍等等。我们希望这一点在我们的模拟中能够有所体现，所以我们考虑构造一个和我们设计的 Markov 链的概率矩阵大小相同的稀疏上三角矩阵，通过生成随机数来模拟我们的关联规则，最后将两个矩阵相加再进行归一化，得到的矩阵就是一个能体现关联规则的模型。此外我们假设如果在某一状态 i 下只有 $P_{i0} \neq 0$，则取 $P_{i0} = 1$，否则取 $P_{i0} = c$，即 c_i 为常数 c。也就是说我们认为每次购买一个商品以后不再选择下一个物品的概率固定。

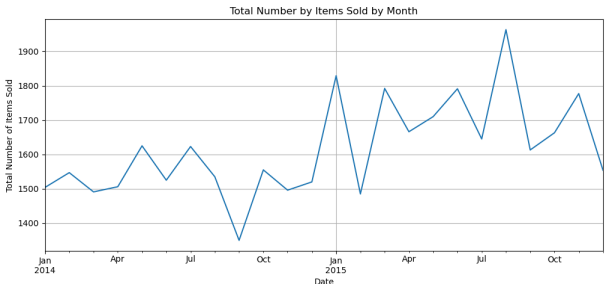
在经过调参后，我们对该 Markov 链进行了两万次游走，得到了两万条路径，模拟两万次购买记录。为了进行比较，我们设计了前文提到的各种概率分布下的 Markov 矩阵，包括全体 P_{ij} 服从均匀分布、正态分布、指数分布、泊松分布等，探究哪一个模型更符合消费者的购物习惯。

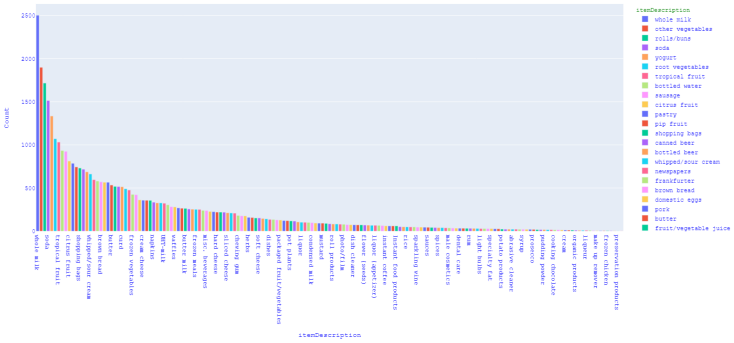
6 结果与对比

6.1 对 groceries_dataset 的数据分析

在正式开始之前我们先对数据集的其它信息进行统计并展示相关信息。groceries_dataset 是数据挖掘比赛 kaggle 网站的一个很出名的数据集。图 1 展示了一些数据集统计特征，虽然与本次建模任务无关但是十分有趣。

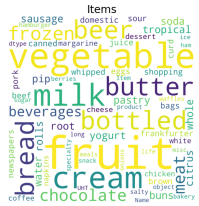






(c) 单个商品的排序

(d) 商品销售总量 treemap 图



(e) 商品销售总量 wordcloud 图

图 1: groceries_dataset 数据统计图

对以上图表进行分析不难发现，商家每日或每周售出的商品在一个范围内波动，且不同商品的销量有差异；在该数据集中 whole milk 销量最高。

6.2 关联规则挖掘

我们先对 groceries_dataset 分别使用 apriori 算法进行关联规则分析。将运行结果整理成表格如下：

	support	itemsets	length
0	0.004010	(Instant food products)	1
1	0.021386	(UHT-milk)	1
2	0.001470	(abrasive cleaner)	1
3	0.001938	(artif. sweetener)	1
⋮	⋮	⋮	⋮
746	0.001002	(rolls/buns, soda, whole milk)	3
747	0.001337	(rolls/buns, whole milk, yogurt)	3
748	0.001069	(whole milk, soda, sausage)	3
749	0.001470	(whole milk, sausage, yogurt)	3

表 2: 使用 apriori 算法挖掘 support≥0.001 的频繁项

	support	itemsets
0	0.157923	(whole milk)
1	0.085879	(yogurt)
2	0.060349	(sausage)
3	0.009490	(semi-finished bread)
4	0.051728	(pastry)
⋮	⋮	⋮
745	0.001403	(yogurt, chewing gum)
746	0.001069	(other vegetables, chewing gum)
747	0.001002	(chewing gum, soda)
748	0.001069	(whole milk, pasta)
749	0.001002	(rolls/buns, seasonal products)

表 3: 使用 fp-growth 算法挖掘 support≥0.001 的频繁项

	antecedents	consequents	···	leverage	conviction
0	(UHT-milk)	(bottled water)	···	-0.000228	0.988755
1	(bottled water)	(UHT-milk)	···	-0.000228	0.996168
2	(other vegetables)	(UHT-milk)	···	-0.000473	0.996060
3	(UHT-milk)	(other vegetables)	···	-0.000473	0.975443
4	(rolls/buns)	(UHT-milk)	···	-0.000548	0.994934

表 4: 使用 apriori 算法挖掘关联规则，上述表格展示了前五条规则

	antecedents	consequents	···	leverage	conviction
32	(yogurt, sausage)	(whole milk)	···	0.000563	1.131541
36	(rolls/buns, sausage)	(whole milk)	···	0.000292	1.069304
38	(soda, sausage)	(whole milk)	···	0.000130	1.026642
40	(semi-finished bread)	(whole milk)	···	0.000172	1.022008
13	(rolls/buns, yogurt)	(whole milk)	···	0.000102	1.015701
⋮	⋮	⋮	⋱	⋮	⋮
580	(tropical fruit)	(chicken)	···	-0.000485	0.992690
919	(pastry)	(berries)	···	-0.000058	0.998861
305	(soda)	(frozen vegetables)	···	-0.000714	0.992489
842	(yogurt)	(napkins)	···	-0.000162	0.998073
458	(citrus fruit)	(chocolate)	···	-0.000184	0.996463

表 5: 使用 fp-growth 算法挖掘关联规则，上述表格展示了前五条与后五条规则

从以上表格可以看出，两种算法最后生成的频繁项集和关联规则相同，但是两种算法的挖掘过程不一样。我们从频繁项集的运行结果就可以看出，apriori 算法先处理较少项集再处理较大项集，而 fp-growth 在生成时先生成频繁项后生成非频繁项。我们还可以对两种算法的运行效率进行对比。从图 2 中可以看出，fp-growth 算法的挖掘速度明显大于 apriori 算法。

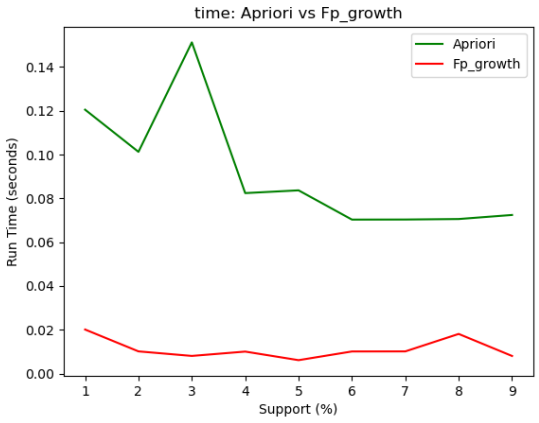


图 2: 两种算法处理相同任务的时间对比

除了算法运行过程以外，我们更关注与数据挖掘有关的几种指标，它们之间的联系可能对判断数据合理性有用。我们作出 support，confidence，lift 三者的关系图如下：

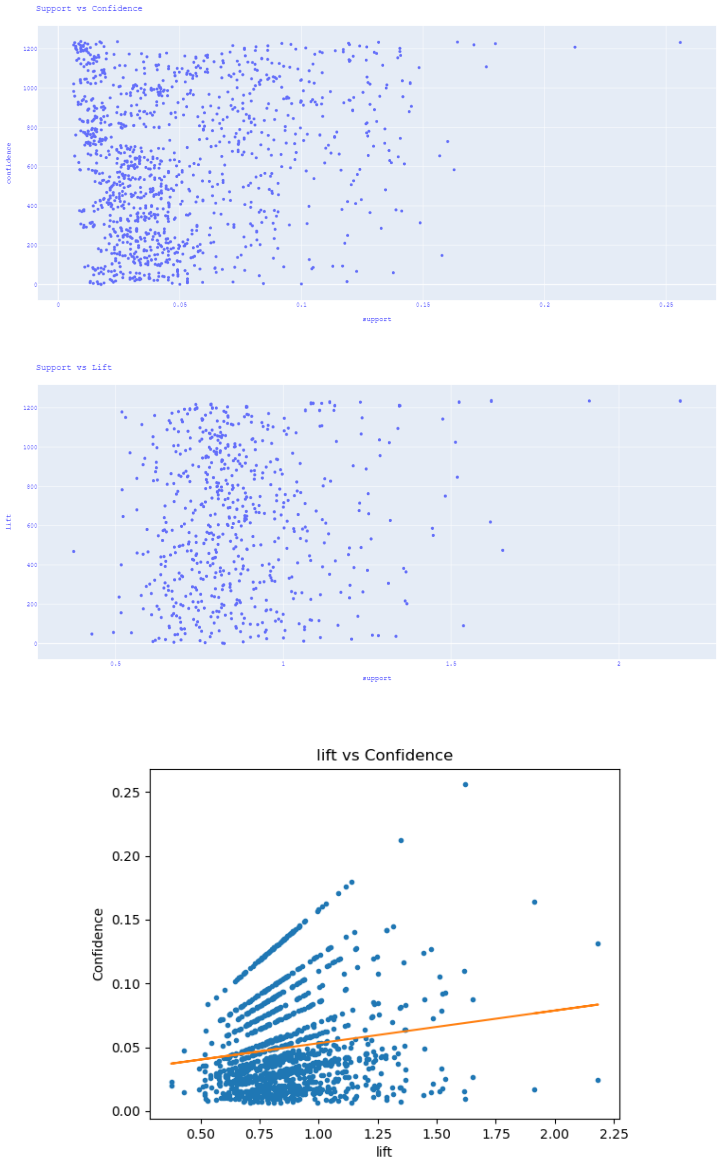


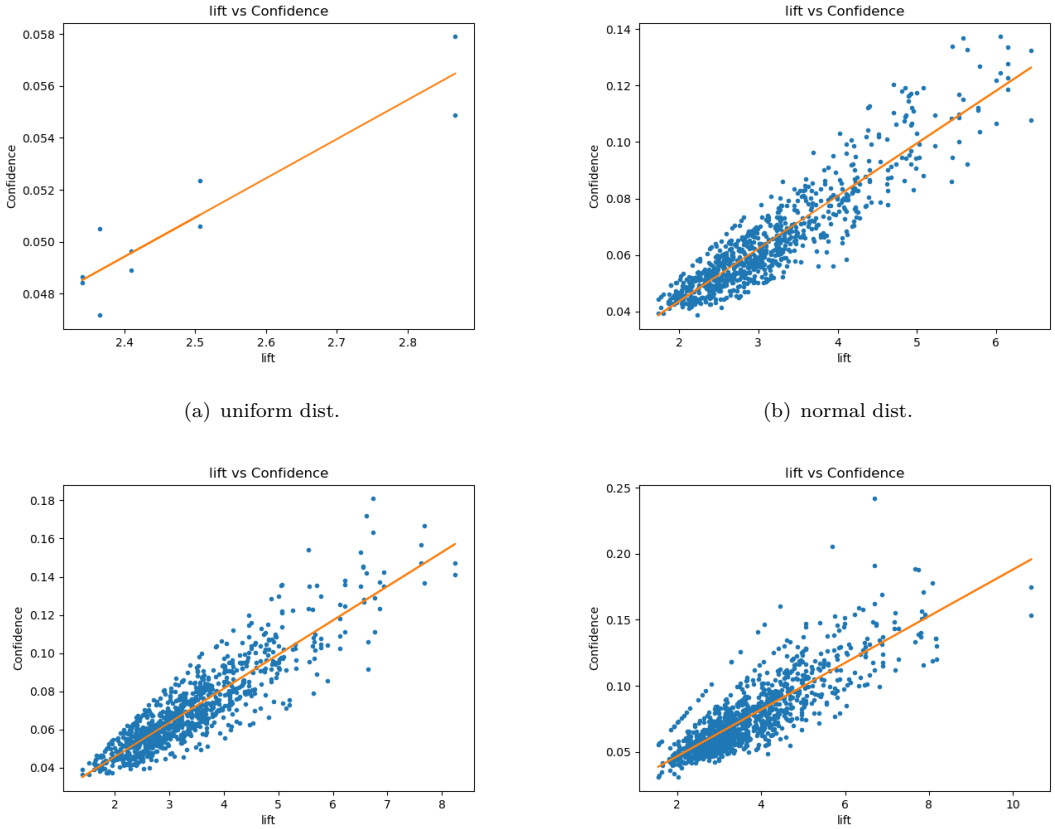
图 3: support,confidence,lift 三者的关系

我们可以看到，support-confidence 图和 support-lift 图没有明显的特征，数据点分布比较杂乱。但是 lift-confidence 图十分“有趣”——它是由一群直线簇组成的。直观上来看，直线簇中每一条直线可能代表了一条关联规则，直线簇可能是关联信息的体现。于是在下面的评价生成数据集合理性时，我们便可以作出每一个数据集对应的 lift-confidence 图，从图中便可判断商品关联性如何。

6.3 消费者购买行为的模拟

在数学模型建立部分中，我们提到了消费者购买行为的生成方法是：将服从某个分布的 Markov 矩阵和为引入关联规则的系数矩阵的叠加。我们暂时不引入关联规则的系数矩阵项，只考虑 Markov 矩阵

生成数据。假定消费者选择某一件商品的概率服从均匀分布、正态分布、指数分布和泊松分布。使用这些分布对应的 Markov 矩阵生成购买数据，对它们绘制 lift-confidence 图（见图 4）以验证商品关联性如下。



(a) uniform dist.

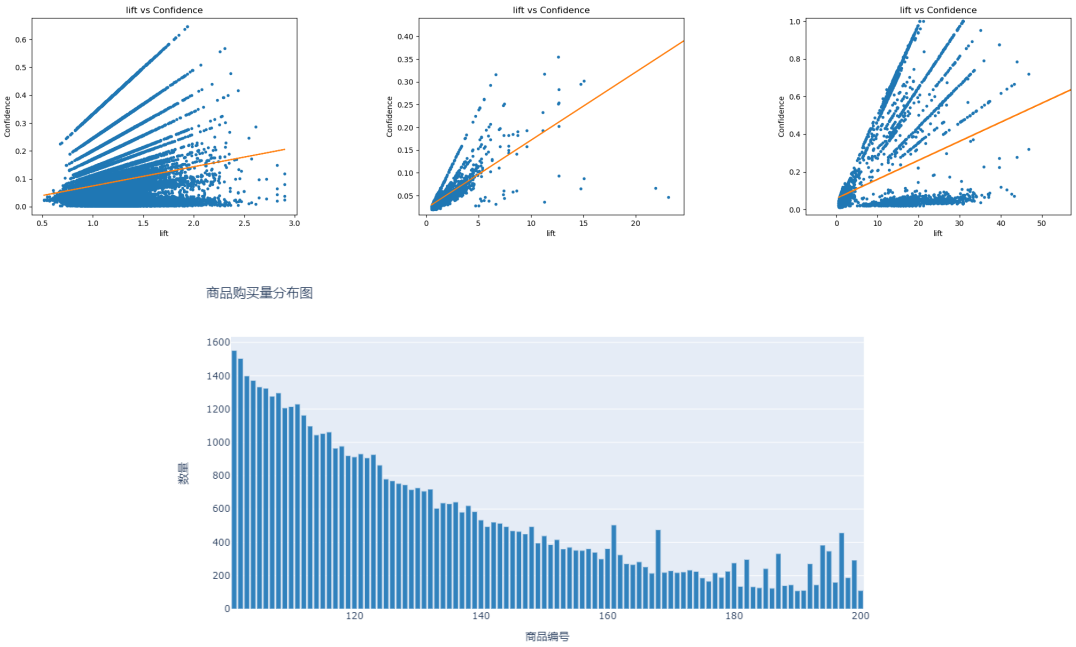
(b) normal dist.

(c) exp dist.

(d) possion dist.

图 4: 四种分布假设下的 lift-confidence 图

不难看出，均匀分布下生成的图像与实际情况差距最大，而正态分布、指数分布与泊松分布下生成的 lift-confidence 图大体上呈线性，与实际情况较相符。我们以各商品的需求有指数分布形式为例：引入了产生关联规则的稀疏项，利用该分布进行了多次重复实验，得到不同数据集，作出了如图 5 所示几组不同的图像。



(d) 其中一次实验生成的数据分布

图 5: 重复实验生成数据集的 lift-confidence 与分布图

我们可以看出，在引入产生关联规则的稀疏项过后，得到的各商品销量分布变得不均匀，与 groceries_dataset 相符，并且 lift-confidence 图中的直线簇明显增多，说明该方法能够有效地引入关联规则，从而为更深层次的研究提供数据。

6.4 商家销售策略的建议

1) 进货策略优化。

通过上面关联规则的挖掘，可以得到相关的商品。这些商品消费者购买时大多数情况会一起买。例如在 DVD 租赁平台时，租借变形金刚 1 的会员大多还会借变形金刚 2。通过这种规则分析，就可以为商家进货提供一种策略：关联规则的后件的进货数量不能少于前件的 90%(或其他值，主要取决于置信度)。在上例中，即变形金刚 2 的 DVD 数量不能少于变形金刚 1 的 DVD 数量。有了这种策略，商家就不容易出现缺货的情况，能满足大部分消费者的消费需求。

2) 捆绑销售。

对于相关商品，消费者购买前件后会有很大概率购买后件。商家就可以直接将前后件捆绑销售，这样用户就可以觉得非常省心，可以一次购入自己想要的全部东西。再加上一些捆绑销售时打折促销等手段，能大大提高消费者的消费满意度。顾客满意自然销量也能大幅增加，能牟取更多利益。

3) 精准营销，精确推荐。

对于在线购物平台，往往会有商品推荐。商品推荐得好，能让消费者消费增加，消费者能买到自己想要的商品；相反，推荐得不好，会增加消费者的反感，不仅花费了时间精力做了推荐系统还让消费者体验不好。所以推荐系统十分重要。通过关联规则分析，能找到相关商品，在消费者购买前件

后推荐后件。这样的推荐方法就大概率能满足消费者的购买想法，销量增加的同时，消费者满意度也会提高。

4) 出售冷门商品。

对于与其他商品关联度低的商品，同时这些商品如果销量很差，则说明这些商品属于冷门商品且不会对其他商品的销量有过多影响。对于商家而言，这些商品就没有很大的价值，商家可以通过打折，赠送等其他方法将已有存货尽快出售，并在下次进货时不再购入这些冷门商品。这样可以避免购入过多的冷门商品影响销量。同时，若只已销量进行评价指标，卖得少的商品进货就少进，这些商品可能能关联一些销量高的商品，这就会影响销量高的商品的销量。例如某部影片第一部的销量因某些特殊原因很差但第二部销量很好，如果直接不购入第一部，那可能第二部的销量也会有所影响，因为大多数新用户还是从第一部开始看起。通过关联分析可以找到真正的" 垃圾商品"，能帮助商家获得更多的利润。

7 结论

本文主要阐述了消费者购物特征中存在关联规则的机理，通过实现两种算法研究并验证了关联规则的假设，并引入 Markov 链模型解决了人类购买行为模型的建立，解决了数据的模拟问题，推动了关联规则的进一步研究。我们发现 apriori 算法与 fp-growth 算法都能够有效生成关联规则，且 fp-growth 算法的效率比 apriori 算法效率更高，同时我们通过假设商品热度呈泊松分布和引入噪声模拟关联的方法有效地对购物数据集进行模拟生成。

8 问题

(1) 我们提出的 Markov 模型假设过于简单，事实上，如果仅考虑最普通的 Markov 链，该 Markov 链的概率矩阵为 A，如果能够构造出一个其平稳分布 $\pi = (\pi_0, \pi_1, \cdots, \pi_{100})$ 满足其服从指数分布，并且 $\pi = \pi A$，那么由 Markov 链的性质，该 Markov 链长期而言处于状态 i 的时间占总的时间的平均比例就是 π_i，也就自然满足了商品的数量分布。但是由于构造该 Markov 链条件过于苛刻，且并不能解出一个通式解，所以我们并不能简单实现。

(2) 本文中只介绍并使用了 apriori 算法以及 FP-tree 两种算法，然而关联规则算法仍有很多如 PCY 算法和多哈希算法，并且这文中这两个算法也有了优化过后的算法如 GPApriori 算法以及 XFP-Tree 算法，其效率会进一步的提升。

参考文献

[1] Pang-Ning Tan, Michael Steinbach, Vipin Kumar. 数据挖掘导论. 北京：人民邮电出版社，2006.

[2] Sheldon M. Ross. 随机过程. 北京：机械工业出版社，2013.