|  |
| --- |
| 江 西 理 工 大 学  本 科 毕 业 设 计（论文）  题 目：基于商品影响力的购物篮分析研究  学 院：信息工程学院  专 业：计算机科学与技术(软件方向)  班 级：121班  学 生：袁国文  学 号：20123151  指导教师：杨书新 职称：副教授    时间：2016年6月1日 |

**摘 要**

购物篮数据反映了购物者的购买历史。购物篮分析可以从购买历史中得到潜在的有价值的信息，这些信息能够辅助零售商选择促销商品。

在对研究现状分析的基础上，本文提出了两个基于商品影响的购物篮分析方法来求解选择促销商品的问题，具体工作如下：

(1)使用apriori算法求取商品之间的关联性，并构建有向带权商品网络。这个商品网络较无向网络可以携带更多的商品关联信息。

(2)侧重考虑商品之间的影响是相互独立，且刚加入购物篮的的商品会让购物者考虑购买相关商品，提出了基于独立级联模型的购物篮分析算法。

(3)侧重考虑商品之间的影响是可以累积的，且商品影响的传播受到购物者考虑这些商品的时间的影响，提出了基于热量传播模型的购物篮分析算法。

(4)以真实的交易数据为基础，通过实验验证了算法的有效性。

关键字**：**购物篮分析；关联规则；商品网络；社团划分；影响最大化

**ABSTRACT**

The market basket data reflects the shopper's purchase history. The market basket analysis can mining the potential valuable information from the purchase history, these information can assist the retailer to choose the promotional merchandise.

Based on investigating domestic and foreign research, this paper proposes two algorithms of market basket analysis based on the influence of commodity to get the promotion commodity.

The main work is as the following:

(1) Construct directed commodity network with weighted of edge. The weight of edge between two commodities is defined as the correlation of them, which can be computed by using apriori algorithm to obtain the correlation between the goods. More information are carried in the network than the undirected network.

(2) In view of influence between the two commodities, a commodity can influence shopper to buy relevant commodities. Under the assumation that the influence is independent, a market basket analysis algorithm based on the independent cascade model is proposed.

(3) Considering that the impact of commodity is cumulative and the influence is affected by shopper as the time goes, we propose a market basket analysis algorithm based on the heat diffusion model.

(4) Conduct experiments on the real market basket data, and the results show the rationality of the proposed algorithms.

**Key words:** market basket analysis; commodity network; community division; influence maximization algorithm

目 录

[第一章 绪论 1](#_Toc453765409)

[1.1研究背景与意义 1](#_Toc453765410)

[1.2国内外研究现状 2](#_Toc453765411)

[1.2.1基于商品网络的购物篮分析的研究现状 2](#_Toc453765412)

[1.2.2社交网络影响最大化的研究现状 3](#_Toc453765413)

[1.3本文的主要工作 4](#_Toc453765414)

[1.4本文组织结构 4](#_Toc453765415)

[第二章 基于商品网络的购物篮分析 6](#_Toc453765416)

[2.1引言 6](#_Toc453765417)

[2.2 apriori算法 6](#_Toc453765418)

[2.2.1定义和性质 7](#_Toc453765419)

[2.2.2计算过程 8](#_Toc453765420)

[2.2.3减枝思想 8](#_Toc453765421)

[2.3构建商品网络的方法 9](#_Toc453765422)

[2.4商品社团发现 11](#_Toc453765423)

[2.5重要商品的发现 11](#_Toc453765424)

[2.6本章小结 12](#_Toc453765425)

[第三章 社交网络影响最大化 13](#_Toc453765426)

[3.1引言 13](#_Toc453765427)

[3.2预备知识 13](#_Toc453765428)

[3.2.1社交网络概述 13](#_Toc453765429)

[3.2.2影响最大化问题的定义 14](#_Toc453765430)

[3.3影响力传播建模 14](#_Toc453765431)

[3.3.1线性阈值模型 14](#_Toc453765432)

[3.3.2独立级联模型 16](#_Toc453765433)

[3.3.3热量传播模型 18](#_Toc453765434)

[3.3.4其他传播模型 19](#_Toc453765435)

[3.4影响最大化关键技术 20](#_Toc453765436)

[3.4.1影响最大化问题求解算法概述 20](#_Toc453765437)

[3.4.2有向图快速层次聚类算法 21](#_Toc453765438)

[3.5本章小结 23](#_Toc453765439)

[第四章 基于商品影响力的购物篮分析 24](#_Toc453765440)

[4.1引言 24](#_Toc453765441)

[4.2商品网络的构建 24](#_Toc453765442)

[4.3基于独立级联模型的购物篮分析 27](#_Toc453765443)

[4.4基于热量传播模型的购物篮分析 28](#_Toc453765444)

[4.5基于社区划分的影响最大化算法 30](#_Toc453765445)

[4.6本章小结 34](#_Toc453765446)

[第五章 实验 35](#_Toc453765447)

[5.1引言 35](#_Toc453765448)

[5.2实验背景 35](#_Toc453765449)

[5.2.1实验环境 35](#_Toc453765450)

[5.2.2实验数据集 35](#_Toc453765451)

[5.2.3商品特征的选择 36](#_Toc453765452)

[5.3实验结果 37](#_Toc453765453)

[5.3.1社区划分结果 37](#_Toc453765454)

[5.3.2基于独立级联模型的购物篮分析的实验结果 39](#_Toc453765455)

[5.3.3基于热量传播模型的购物篮分析的实验结果 40](#_Toc453765456)

[5.3.4对比简析 42](#_Toc453765457)

[5.5本章小结 42](#_Toc453765458)

[第六章 总结与展望 43](#_Toc453765459)

[附 录 44](#_Toc453765460)

[独立级联模型的主要代码 44](#_Toc453765461)

[热量传播模型的主要代码 47](#_Toc453765462)

[社区划分的主要代码 51](#_Toc453765463)

[参考文献 55](#_Toc453765464)

[外文资料 57](#_Toc453765465)

[外文文献1和译文 57](#_Toc453765466)

[外文文献2和译文 61](#_Toc453765467)

[外文文献3和译文 62](#_Toc453765468)

[外文文献4和译文 63](#_Toc453765469)

[致 谢 65](#_Toc453765470)

# 第一章 绪论

随着经济水平的提高，人们在闲暇时间逛逛超市，不管是商品的购买量和光顾超市的频率都在快速增加，人们的需求带动了超市的发展。我们很容易想到这样的一个场景，超市进行着促销活动，在pos机前人们手中提着购物篮排着长长的队伍，收银员手忙脚乱的录入着顾客购物篮中商品的信息。在超市经营中积累的大量的这些数据叫做购物篮数据。

在居民区总能方便的找到超市，这也意味着人们可选择的超市在增多，超市的竞争压力也在增加。因此零售商们想着各种方法来提高顾客的购物体验，其中一种常用的方法就是通过分析购物篮数据来刻画顾客的购物行为，然后零售商再根据顾客的特点来制定策略改善顾客体验。

购物篮分析的英文叫做Market Basket Analysis，简称为MBA。商品关联性分析是MBA的一个重要的组成部分，其中一个经典的故事是啤酒和尿布的故事。这个故事讲述的是美国年轻父亲在周五上下班期间会在沃尔玛超市中同时购买尿布和啤酒这两样很难联系到一起的商品。这个现像在20世纪90年代被美国沃尔玛超市发现，超市通过调整尿布和啤酒的布局带来了尿布和啤酒销量的提升。

啤酒和尿布的故事体现了MBA研究的商业价值，MBA的研究有很多，但是基于商品网络来进行MBA研究仍比较少，本文将利用apriori算法求取商品关联性和构建商品网络，并新颖的用社交网络的影响最大化算法从商品网络寻找影响力延展度最大的前*k*件商品，进而求解选择促销商品的问题。

## 1.1研究背景与意义

MBA是一种广泛使用的技术，可以用来分析顾客同时购买某些商品或类别的可能性。1992年，Julander首次提出购物篮分析[[[1]](#endnote-1)]方法，该方法被瑞典的一个超市所使用。1993年Agrawal等人[[[2]](#endnote-2)]提出了在购物篮发现关联规则的算法，这个算法利用条件概率的思想，从零售数据中发现潜在关联规则，这个技术后来被很多研究者所使用和优化。这个技术典型的应用是评估零售店的商品关联性。基于数据挖掘的思想，购物篮分析的目的是通过从pos机上的交易记录发现顾客购买商品的关联性 [[[3]](#endnote-3)] [[[4]](#endnote-4)]。或许最著名的关联规则是啤酒和尿布，因为家庭在周末到来时候经常同时购买购买啤酒和婴儿尿布。关联规则分析还被应用到很多的场景中，如cross-selling，辅助决策，顾客行为分析和客户关系管理 [[[5]](#endnote-5)] [[[6]](#endnote-6)]。在2010年，Ting和Steve等人[[[7]](#endnote-7)]将购物篮的关联分析应用到了餐馆的菜单选择上。

MBA的目的是利用商品的关联性诱导客户购买非计划外的商品。2009年Bell等人[[[8]](#endnote-8)]发现非计划购买不是一种常见现象。顾客购买行为更多的以来于用户的计划和习惯，还有收集的商品信息。关联性本身可能与直觉相反，例如MBA方法有时错误的辨别互补商品（顾客喜欢一起购买的商品）或者是替代品（多购买一个可能是为了迎合别人的意愿）。2011年Bogdan Hoanca等人[[[9]](#endnote-9)]用MBA方法根据顾客的“价格敏感度和关联的饱和度”来估算一个大学小书店的潜在收入增长。同一年Chrndra等人[[[10]](#endnote-10)]研究如何设计一个好的样本子集来进行购物篮分析，因为在交易量巨大的企业中很难直接在整个数据集合采用购物篮分析。2013年Wiley Interdisc等人[[[11]](#endnote-11)]使用MapReduce来提高购物篮分析效率。

因商品网络的直观特性，如商品网络的边能直观的反应商品之间的相关性，还能利用可视化的技术和图论的知识站在整个网络的角度进行购物篮分析。2012年Hyea Kyeong Kim等人[[[12]](#endnote-12)]构建了商品网络并分析商品网络的边权值分布，节点的中心度数与热销的关系等。通过了解近几年购物篮分析技术的发展，利用商品网络来进行购物篮分析的研究仍比较少，并且只是对商品网络的简单性质进行分析，如度数分布等。

利用商品网络进行购物篮分析因其研究仍比较少，所以存在很大的探究空间。在本文，为了更深入的对购物篮进行分析。在商品网络基础上，新颖的利用社交网络影响最大化方法来寻找延展度最大的前*k*个商品。商品影响力的延展度在一定程度上反映了这些商品如果作为促销商品，它们可能在全局性的带动其它商品的销售。因此本文的研究可以辅助零售商进行商品促销的决策。

## 1.2国内外研究现状

因基于影响最大化的购物篮分析，研究较少，所以国内外现状分成两部分阐述，一是基于商品网络的购物篮分析问题的研究现状。二是社交网络影响最大化问题的研究现状。

### 1.2.1基于商品网络的购物篮分析的研究现状

研究者渐渐意识到商品网络所具有的直观性质，商品网络的节点可以用来表示商品，而边可以用来表示商品之间的关联性。复杂网络理论研究目前是较为成熟的理论，能够给商品网络的研究提供很多的理论支持。此外，复杂网络的可视化技术能够给商品网络分析带来很多的便利。

2012年Hyea Kyeong Kim等人[12]为了能从整体的视角上来分析网络的，构建了商品网络。商品网络的节点代表商品，节点之间的边代表商品一起被购买过。利用了关联规则算法求得的商品之间的关联性作为商品之间的边的权重，边权重越大则代表商品之间关联性越强。在该商品网络上，Hyea Kyeong Kim等人[12]对商品网络的基本属性进行了分析，如边权分布，节点度数分布以及节点的degree centrality与商品热销程度的关系等。

2014年戚戚[[[13]](#endnote-13)]将一起购买的商品均认为存在边的关系，构建起了无权值无向图。戚戚在无向图上使用层次聚类进行了社团研究，并借助igraph挖掘社团中的核心商品。除此，为了在商品网络中给商品的重要性排序，戚戚将无向边转化为两条方向相反的有向边，边权值均取值为1，将无向图转化为有向带权图，并使用pagerank算法挖掘重要的商品。

2015年Zhu Z等人[[[14]](#endnote-14)]以周为时间窗口，寻找出用户每隔一段时间会购买的商品组合。根据这些商品组合，他们认为在同一组的商品存在边的关系，由此构建了一个无向的商品网络。为了刻画用户的消费行为，他们提出从无向的商品网络自动抽取商品小类的方法。

基于商品网络的购物篮分析的研究仍比较少，从2012年Hyea Kyeong Kim等人[12]对商品网络的进行简单的度数统计到2015年Zhu Z等人[14]的抽取商品小类别。基于商品网络的购物篮分析方法日趋丰富，但仍有很多的发展空间。

### 1.2.2社交网络影响最大化的研究现状

在社会生活中，个体之间都存在相互影响的关系，一个人的偏好会受到其朋友，家人等的影响，同时也会影响到其他人。比如对某件商品的评价，可能会影响其他人对该商品的评价，同样对音乐的评价，餐馆的评价等也存在这种影响的传播，在社交网络中叫做口碑传播，也叫做影响力的传播。

深入理解人类的行为和影响力的传播，商家可以给顾客提供更加个性化和舒适的服务，及迅速提高品牌价值。政府可以根据影响力的传播特性，迅速找到部分高影响力的人群快速推广政策，也可以截断高影响传播路径迅速控制谣言的传播等。

影响力传播研究有三大支柱，第一是影响力传播模型，主要是分析影响力在社交网络中传播的特点。第二是影响力传播学习，即根据机器学习等方法挖掘出影响力的传播模型和模型的参数。第三是影响力传播优化，即寻找方法来增大想要传播的影响，或者减小不想传播的影响。

影响最大化（influence maximization）问题一个被广泛研究的影响力传播优化问题。影响最大化是指在网络中寻找*k*个节点，这*k*个节点能使影响力的延展度最大，所以该问题是一个组合最优化的问题，在某些模型下（独立级联模型，线性阈值模型），是一个NP难的问题。

2003年，Kempe等人[[[15]](#endnote-15)]指出在独立级联模型和线性阈值模型下,影响最大化节点集合的选取是一个NP难问题。因此Kempe等人使用贪心算法去求解该问题的近似解。2007年Leskovec J等人[[[16]](#endnote-16)]发现在独立级联模型和线性阈值模型下，影响最大化问题呈现出很好的子模特性，在使用贪心算法每次选取影响最大的种子节点的时候没必要更新所有节点的影响力，由此提出了给节点增加影响力更新标识的CELF算法。该算法减少了节点影响力更新的次数，在一定程度上提高了贪心算法的效率。2010年wei chen等人[[[17]](#endnote-17)]发现独立级联模型可用随机级联模型来模拟，通过随机移除边，然后计算节点的可达集合大小来代表节点的影响力。2012年Borgs C等人[[[18]](#endnote-18)]提出了反向蒙特卡洛算法，该方法能在近似线性时间求解出影响力最大的种子集合，同时仍然有()的近似比保证。2013年，Hui Li等人[[[19]](#endnote-19)]提出了基于Conformity-Aware的贪心算法，该算法先使用社区划分的方法进行社区划分，然后对所有社区进行种子集合的选取。

除了基于独立级联模型和线性阈值模型的影响最大化算法， 2008年Hao Ma等人[[[20]](#endnote-20)]认为热量在传播媒介上随时间的扩散与影响力的传播相似，因此创新性的提出使用热量传播模型（Heat Diffuse Model， HDM）来模拟复杂的影响力传播。该方法求节点的延展度时候，不需要进行多次的影响力传播模拟并取平均值来近似。

2014年Chen等人[[[21]](#endnote-21)]为了缩小影响力种子候选集合的大小以及避免影响力种子集中于一个社区，提出了适用于无向图的基于层次聚类的影响最大化方法，该方法分为三个步骤，第一社区发现，第二选取候选集合，第三种子集合的选取。

## 1.3本文的主要工作

基于商品网络的购物篮分析是近几年来出现的购物篮分析方法，国内外对该方法的研究仍比较少。图论的研究已经很成熟了，近年来也涌现大量的复杂网络数据挖掘的方法。在商品网络中的应用，缺乏实际情况的检验，所以很多的研究仍处于探究阶段。使用社交网络影响最大化方法在商品网络中寻找影响最大的种子集合的研究仍比较少，本文从探究的角度进行了如下的工作：

（1）搜集基于基于商品网络的购物篮分析的方法和社交网络的影响最大化的方法，厘清这两种研究的发展趋势和现状，了解主流方法的求解思路和特点，并选取合适的方法来进行本文的探究性实验。

（2）在选取商品影响力最大的种子集合时候，为了解决商品集中在一个社

区的问题，本文采用了社团划分的方法，进行种子选取的时候可根据社区种子的选取情况，对社区的种子选取进行控制。

（3）本文选取独立级联模型和热量传播模型对商品影响传播进行建模，并设计和实现了基于商品影响力的购物篮分析方法。

（4）在真实数据中分别执行了基于独立级联模型和热量传播模型的购物篮分析算法，并对实验结果进行分析。

## 1.4本文组织结构

本文的正文本分为六章，具体组织结构如下：

第一章为绪论。该章先用一个例子点出了购物篮分析的商业价值，然后介绍购物篮分析的背景以及本文的研究意义，接着总结本文所用理论的国内外研究现状，最后阐述本文的主要研究内容和组织结构

第二章为基于商品网络的购物篮分析。该章介绍购物篮的定义和购物篮分析的应用研究，然后介绍了购物篮分析中常用于分析商品关联性的apriori算法，为构建商品网络做一个知识铺垫。然后介绍了基于商品网络的购物篮分析的相关知识，如构建商品网络的方法，基于商品网络的商品社区发现，重要商品发现等。

第三章为社交网络影响最大化。该章介绍本研究中会用到的社交网络的基础知识，包括社交网络的定义，影响最大化问题的定义，影响力传播模型，以及求解影响最大化问题的关键技术。

第四章为基于商品影响力的购物篮分析。本章以商品网络为衔接点，将社交网络影响力最大化算法用于购物篮分析中，提出了基于商品影响力的购物篮分析方法。该方法分为三个步骤求解选择促销商品的问题。第一步是基于apriori算法的思想求取商品间的相关性，并构建商品网络。第二步是对商品影响进行建模，分别基于独立级联模型和热量传播模型实现了求影响增益的算法。第三步是设计基于影响传播模型的购物篮分析算法，该算法采用有向图的快速层次聚类算法来进行社区划分，并对每个社区进行计分，用于选取影响潜力更大的商品。

第五章为实验。该章主要是介绍实验的背景，以及对实验结果进行分析。

第六章为总结与展望。总结本文的内容，包括研究问题的来源，研究过程，收获以及未来的计划。

# 第二章 基于商品网络的购物篮分析

## 2.1引言

随着经济水平的提高，因为人们的需求，超市的规模和数量都在增加。随着顾客可选的超市数量增加，超市的激烈越发激烈。超市对顾客消费行为的了解程度是超市在激烈竞争中胜出的关键因素。自从1992年Julander等人[1]提出购物篮分析方法，购物篮分析的研究便开始兴起。

购物篮是超市中供顾客装商品的一种篮子。在顾客付款的时候，收银员会将篮子中的商品信息会录入到POS机器中，一个客户一次购买的所有商品的流水号是相同的。购物篮分析是通过分析这些交易记录来发现顾客的消费行为。购物篮分析在零售行业的应用[[[22]](#endnote-22)]主要有：

（1）商品的布局设计。通过对购物篮的数据进行商品的关联性分析可以知道顾客在购买一些商品之后喜欢购买哪些商品。那么零售商可根据的顾客喜好进行商品的布局的调整。

（2）促销活动的有效性分析。从购物篮数据中分析以何种策略进行商品的促销能带来更多的回报，包括促销商品的选择，促销价格的设置，促销时间长度设置等。

（3）顾客分类。根据顾客的消费特征对顾客进行分类管理，可以更有针对性的为顾客的推荐服务。

（4）顾客的忠诚度分析。顾客的会员卡记录者使用者的购买行为，通过时间序列上的购买分析可以了解顾客的消费喜好的变化和忠诚度的变化，那么零售商可以以此为依据制定策略留住老顾客。

（5）交叉销售。通过分析交易记录了解顾客的多种需求，在满足顾客的某一个需求的时候同时尝试去推荐相关的商品满足相关的需求。比如在满足顾客购买手机的需求下，尝试去满足顾客对屏幕保护的需求。

（6）降低库存成本。通过对超市的交易记录可以了解顾客在某些时间（季度等）对商品的销售量，那么商家可以针对性的调节进货量等。

（7）市场和趋势分析。主要是根据交易信息预测商品的需求量等，可以有针对性的调节商品的价格和库存等。

接下来着重介绍购物篮分析中求取商品相关性的apriori算法，以及其他基于商品网络的购物篮分析的相关知识，如构建商品网络的方法、基于商品网络的商品社区发现和重要商品发现等。

## 2.2 apriori算法

apriori算法是关联规则的一个经典算法，该方法常用于购物篮分析求取商品间的相关性质。在本文中会用到apriori 算法来构建商品网络，所以在这里提前介绍基础的知识。

### 2.2.1定义和性质

（1）定义

apriori有两个阶段，第一阶段是求满足最小支持度的频繁项集，第二阶段是从频繁项集构造满足最小置信度的规则。该算法有四个的关键的信息，支持度（support degree）、置信度（confident degree）、最小支持度（minimize support degree）和最小置信度（minimize confident degree）。为了方便描述算法先引入，文中通用符号对应含义如表2-1所示。

表2-1 airiori算法的符号

|  |  |
| --- | --- |
| 符号 | 含义 |
| *num*(*A*) | 商品*A*在商品*A*中出现的次数 |
| *T* | 交易记录的数量 |
| *support*(*A*) | 商品*A*的支持度 |
| *confident*(*B*|*A*) | 购买*A*商品后购买*B*商品的可能性 |
| *minSupport* | 最小支持度 |
| *minConfident* | 最小置信度 |
|  | 表示集合大小为1的商品集合，称为1项集 |
|  | 表示集合大小为*k*的商品集合，称为*k*项集 |

（1）支持度（support degree）表示商品或商品组合在交易记录中出现的频率。可用公式（2.1）计算商品*A*的支持度：

（2.1）

（2）置信度（confident degree）表示购买一些商品后购买另外一些商品的可能性。已经知道商品*A*，*B*，{*A,B*}的支持度，可用公式（2.2）求商品*A*后购买商品的*B*的置信度：

(2.2)

（3）最小支持度（minimize support degree）和最小置信度（minimize confident degree）是为了提高算法效率和约束求解结果设置的。大于等于最小支持度的项集被称为频繁项集合。

（2）重要性质

（1）如果*S*是频繁项集，那么*S*的任何一个子集也是频繁项集。

（2）如果*S*是非频繁项集合，那么*S*的任何超集也是非频繁项集。

### 2.2.2计算过程

（1）支持度计算

计算项集的支持度的过程如表2-2：

表2-2 计算支持度的过程

|  |
| --- |
| （1）计算含有一个元素的项集出现的频率，然后筛选出的项集记为。  （2）将两两组合得到集合大小为*k*的项集合，并求，然后筛选的项集记为。循环处理直到没有更大的频繁项集合产生。 |

（2）关联规则计算

对于每一个频繁项集，将的集合大小为*m*的子集记，记。

计算项集间的关联性的过程如表2-3：

表2-3 计算项集间的关联性的过程

|  |
| --- |
| 对每一个  （1）根据公式（2.2）计算，然后筛选出的项集记为。  （2）将两两组合构成集合大小为2的，计算，然后筛选出的项集记为。重复这个步骤直到*n == k*。 |

### 2.2.3减枝思想

（1）计算项集的支持度剪枝思想

求项集利用*minSupport*进行剪枝的过程如图2-1所示：如图假设现在交易记录中出现过的商品有商品*A*，*B*，*C*。当求项集的支持度的时候，如果，由式2.1可知含有*B*商品的项集支持度都会小于最小支持度，那么这些项集的支持度都没必要去算了，即如图2-1所示红色的节点{*B,AB,ABC*}剪枝过程会被去掉。



图 2-1 支持度计算的剪枝

（2）计算关联规则剪枝思想

求关联规则的剪枝过程如图2-2所示，*A*，*B*，*C*表示交易记录中出现的商品，在这个例子中假设{*A, B, C*}是频繁项集，，当,则。当求*BC->A*的置信度的时候，如果，由公式（2.2）可知如果， 那么，的置信度就没必要去算了，即如图2-2所示红色的节点会被剪枝去掉。



图2-2 关联规则计算时的剪枝过程

## 2.3构建商品网络的方法

商品网络分析是购物篮分析的一种新的方法，网络的节点表示商品，网络的边表示商品间存在关系，网络的边权表示关系的强度。商品网络可以可以直观的表示商品间的关系，并且能将购物篮分析扩展到网络的分析上。

商品网络常见的构建方法有5种：

(1) 根据购物篮的数据抽取出的商品信息构成表2-4，*A*,*B*,*C*代表商品，如果交易号中出现了该商品则在相应位置填充1，否则填充0。

表2-4 商品网络构建的例子

|  |  |  |  |
| --- | --- | --- | --- |
| **交易号** | ***A*** | ***B*** | ***C*** |
| 1 | 1 | 1 | 1 |
| 2 | 0 | 1 | 1 |

网络1。2014年戚戚[13]等人把商品 作为网络中的节点，如果两个商品在同一条交易记录中出现，那么就认为这两个商品节点之间存在一条边。根据表2-4的信息可构建的无向商品网络如图2-3所示：



图2-3 商品网络1

(2) 网络2。为了使用pagerank算法对商品的重要性进行打分，2014年戚戚[13]等人将商品网络的无向边转化方向相反的有向边，边的权值均置为1。以图2-3为例可转化成图2-4：



图2-4 商品网络2

(3) 网络3。2012年Hyea Kyeong Kim等人[12]以商品作为节点，如果两个商品在同一客户同一个时间被购买，那么就认为这两个商品节点之间存在一条边。然后将每天一起购买的商品当作一个购物篮，利用apriori算法求取商品间的关联性作为商品间边的权值。

（4）网络4。2012年Hyea Kyeong Kim等人[12]使用的另外一种构建购物篮的方法是，将一个用户在一个时间段购买的商品当作是一起购买，然后以购买的商品作为为节点，然后一个购买则认为存在一条无向边，构建起了无向商品网络。

（5）网络5。2015年Zhu Z等人[14]对喜欢一起购买购买的商品定义为顾客每个时间周期（如每周等）会一起购买的商品。按这个要求从原始交易记录进行筛选出交易记录的子集。在这个新数据集上，把商品看作节点，把一起购买的商品建立边，构建起了一个无向商品网络。这个方法与（4）的区别是（4）是以（客户，时间段）作为分组依据，而这里不仅仅是分组还有周期购买的约束。

假设原始交易记录如表2-5，*A*,*B*,*C*代表商品，如果交易号中出现了该商品则在相应位置填充1，否则填充0。

表2-5 交易记录例子

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **交易号** | **顾客** | **时间** | ***A*** | ***B*** | ***C*** |
| 1 | 1 | 20150103 | 1 | 1 | 1 |
| 2 | 2 | 20150103 | 0 | 1 | 1 |
| 3 | 1 | 20150108 | 1 | 0 | 1 |
| 4 | 2 | 20150108 | 1 | 1 | 1 |

假设以7天为时间周期，如表顾客1在2015年01月03日购买了商品*A*,*B*,*C*后，7天内又购买了*A*,*C*，重复购买的商品是*A*,*C*。同理顾客2重复购买的是*B*,*C*。筛选出新的交易信息如表2-6所示，根据表2-6构建的网络如图2-5：

表2-6 购买记录

|  |  |  |
| --- | --- | --- |
| ***A*** | ***B*** | ***C*** |
| 1 | 1 | 0 |
| 0 | 1 | 1 |



图2-5 商品网络3

## 2.4商品社团发现

商品社团的发现算法可以分为两类，分别是非重叠商品社团发现算法和重叠商品社团发现算法。

1、层次聚类。

一种非重叠商品社团发现算法，层次聚类[13]的过程是先将每个节点看作是一个社区，然后选取距离最近（或相似度最大）的两个社区进行合并，当整个网络的模块值最大停止合并。戚戚[13]使用该方法对2.4节的网络1进行社区划分,这些社区可以给商品分组布局提供依据。

2、*k*-团发现。

这是一个重叠商品社团发现算法，可以寻找出网络中的所有*k*个节点的完全图。Zhu Z等人[14]使用该方法对2.4节的网络5进行了*k*-团发现[14]。在对这些*k*-团分析的是否发现，*k*-团中的商品存在互补的关系。如报警器和烟雾探测器是家庭安全设备，它们在同一团中。

## 2.5重要商品的发现

这里介绍两种从购物篮数据中挖掘重要商品的方法，分别是pagerank算法和度中心性分析。

（1）pagerank算法[13]。

PageRank算法的作用是对按重要性对网页排序。pagerank算法的核心思想是一个网页的重要性由两点来决定，一是越多网页的链接指向该网页，那么该网页越重要，即pagerank值越大。二是该页面越重要，那么的链接指向的页面也就越重要。

戚戚[13]在网络2中使用pagerank算法按重要性对商品进行排序，发现排在首位的是时尚杂志。结果具有一定的合理性，顾客休息或等待同伴时候喜欢观看杂志，并且时尚杂志比起其他专业书籍在更多年龄段的人群受欢迎。

（2）度中心性（Degree Centrality）分析[12]。

度中心性是节点中心性最直接的度量，一个节点的度数越大，度中心性也会越大，反映了该商品节点越重要。

2012年Hyea Kyeong Kim等人[12]发现网络3度中心性大的商品是一些购买较频繁的日常必需品，而网络4中度中心性大的商品是普通家庭的非经常否买的生活必须品。

## 2.6本章小结

本章介绍了购物篮的定义和购物篮分析的常见应用，然后介绍了关联规则分析算法中的apriori，包括该算法的定义，核心思想以及执行过程，为下文的商品网络的构建做了知识铺垫。接着以戚戚[13]、Hyea Kyeong Kim等人[12]和Zhu Z等人[14]的商品网络研究为例子的介绍常见商品网络构建方法以及介绍常见的商品网络分析方法，如社区发现和重要商品发现等。

# 第三章 社交网络影响最大化

## 3.1引言

影响最大化问题是社交网络影响力传播研究的一个热点问题。影响最大化是一个组合最优化问题，求解有3个步骤：1、节点的间相互影响强度的度量，2、选取合适的影响力传播模型，3、选择影响最大化求解算法。

因为本文探究社交网络影响最大化算法在商品网络中的应用，所以在本章以求解影响最大化问题的步骤为顺序，预先介绍基础的理论知识，包括社交网络的概述，影响最大化问题的定义，影响力传播建模，影响最大化问题研究的关键技术（包括影响最大化问题求解算法概述和本文求解影响最大化问题用到的有向图层次聚类）。

## 3.2预备知识

### 3.2.1社交网络概述

社会生活中，个体之间会在某种社会关系，信息会通过这些关系进行传播。为了方便的对社交关系进行研究，常用的方法是对社会关系进行网络建模。网络的节点集*V*={,,…, }表示社会中的个体的集合，网络的边的集合表示节点间的关系，表示节点*i*对节点*j*存在某种关系。

在社交网络中，关系存在很多种，相应的网络也有很多种。在本文中将讨论个体间的影响关系，因为个体间的相互影响关系是不同，所以使用一个有向带权的网络来表示。网络可简单的定义为，其中*V*是节点集合，*E*是边的集合，而*W*是边上的权值。表示社会的一个个体，表示个体可影响个体，影响大小用来表示。与存在影响关系的个体我们称为邻居，用来表示。影响节点的节点称为节点入邻居，用表示，入邻居对影响力总和为。相反，被节点影响的节点称为节点出邻居用表示，节点对节点点邻居节点总的影响为。

为了方便进行相关理论的描述，这里先定义通用的符号如表3-1所示：

表3-1 社交网络通用符号的含义

|  |  |
| --- | --- |
| 符号 | 含义 |
| *G*(*V,E,W*) | 社交网络图 |
| *V* | *G*(*V,E,W*)中的个体集合 |
| *n* | *G*(*V,E,W*)的个体数量 |
|  | *G*(*V,E,W*)中第*i*个节点 |
|  | 带权出度 |
|  | 带权入度 |
|  | 带权度 |
|  | 的出度 |

续表3-1 社交网络通用符号的含义

|  |  |
| --- | --- |
|  | 的入度 |
|  | 的度数 |
| *E* | *G*(*V,E,W*)中的个体间关系集合 |
| *M* | *G*(*V,E,W*)中*W*中边权的和 |
|  | 编号为*i*的边 |
|  | 以为起点，为终点的边 |
| *S* | 种子节点集 |
| *­seed\_num* | *S*中所能包含的最大的节点数量 |
|  | 的邻居集合 |
|  | 节点出邻居集合 |
|  | 节点的入邻居集合 |

### 3.2.2影响最大化问题的定义

在给定的网络，影响力传播模型和所需的参数的情况下，每个节点的影响力，即能够成功影响多少的节点是可以通过计算得到的。我们先假设是一个计算影响力的函数，那么集合*S*的影响力可用来表示。

影响最大化问题是从网络中选择*seed\_num*个种子节点的集合，这个种子集合的比其他*seed\_num*个节点组成的集合影响力大。我们可用公式(3.1)来表示：

。 (3.1)

## 3.3影响力传播建模

在影响力传播模型中最常用的传播模型是线性阈值模型和独立级联模型，热量传播模型则跳出了这个两个模型的思维。在该节中将讲述这三个模型的思想，而伪代码只列出与本文研究有关的独立级联模型和热量传播模型伪代码。

### 3.3.1线性阈值模型

2003年，Kempe、Kleinberg等人[15]将线性阈值模型引入到计算机科学中来对影响力传播进行建模。线性阈值模型中用户有两种状态，活跃状态和非活跃状态。活跃状态可以去影响非活跃状态的邻居，并且用户从非活跃变到活跃状态，就会一直保持活跃状态。线性阈值模型的特点是该模型认为影响是可以积累的，且站在被影响者的角度的来考虑被影响程度。线性阈值模型有一个约束，对于，满足不等式(3.2)。

(3.2)

线性阈值模型中每一个节点会分配一直激活阈值，越大说明节点越难被激活。让表示从最初种子集合到第*t*轮已经被激活的节点，表示初始的节点集合，。在第*t*+1轮不活跃节点的活跃入邻居是的，如果能在该*t*+1轮被激活需满足条件(3.3)：

(3.3)

线性阈值模型的过程是每次用活跃节点去激活新节点得到活跃节点，直到不再有新的节点被激活为止。

以图3-1为例子， 该图是一个有向图，含有4个节点和6条边。



图3-1 有向图1

假设边的权值=，那么该图的权值矩阵如表3-1所示，第一行和第一列是节点的编号。并且所有节点激活阈值均设置为0.5。

表3-1 线性阈值联模型中有向图1的权值矩阵

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** |
| **0** | 0 | 0.5 | 0 | 0.5 |
| **1** | 0 | 0 | 0.5 | 0.5 |
| **2** | 0 | 0 | 0 | 0.5 |
| **3** | 0 | 1 | 0 | 0 |

如果以0号节点为种子节点，求0号种子节点的影响力，将当前被激活的节点集合记*A*。首先将0加入*A*，此时*A*={0}，选择*A*集合中节点的一个未被激活的邻居节点，假设选择了1号节点，根据公式3.3可得：

*A*中节点对1号节点的影响力是0.5，等于1号节点的激活阈值0.5,所以1号节点被激活，将1号节点加入*A*中得到*A*={0,1}。然后再选择*A*的一个未激活邻居节点，假设是3。同样根据公式3.3可得：

*A*中节点对3号节点的影响力是1，大于3号节点的激活阈值0.5,所以3号节点被激活，将3号节点加入*A*中得到*A*={0,1,3}。同样的再尝试激活*A*一个邻居节点，此时选中了2号节点，根据公式(3.3)可知2号节点被激活，将2号节点加入到*A*中，此时*A*={0,1,2,3}。从*A*中选取邻居，发现没有邻居可激活了，即都被激活了。所以停止激活。将*A*中的节点数作为*A*的影响力，此时*A*的影响力是4。如果是其它的例子，可能会出现，*A*的邻居存在没有被激活的，且这些邻居不能被*A*所激活，此时也要停止激活。

### 3.3.2独立级联模型

2003年，Kempe、Kleinberg等人[15]提出独立级联和线性阈值模型等离散递进性传播模型和它们的扩展模型。这些模型简单，符合人们对影响力传播的直觉理解，现在成为了影响力传播的经典模型。2009年W. Chen等人[[[23]](#endnote-23)]改进了独立级联模型，通过概率移除边，然后计算从每个节点出发的可达节点数作为相应节点的影响力，为了使结果更准确进行多次实现取平均可达数作为节点的影响力。本次实验将使用这个影响力传播模型。

对于一个给定的有向的带权网络，*V*是节点集，*E*是边集合，是边的权值集合，表示从节点到节点的边的权重。用用于表示节点的入邻居节点，用用于表示节点的出邻居节点。假设预先选好的初始节点集合记为，节点的影响力记作。用表示在*t*轮被激活的节点。在独立级联模型中，第*t*轮()的每个节点对中未被激活的节点尝试以的概率激活一次。未被激活的节点在第*t*轮被激活的概率可用式（3.4）来计算。算法不断使用*t*-1轮的激活的节点尝试去激活未被激活的节点得到*t*轮激活的节点，直到没有新的节点被激活。初始的节点集合的影响力计算如式（3.5）。节点的影响增益的计算如式（3.6）。

(3.4)

(3.5)

(3.6)

节点只会尝试一次通过激活节点，因此我们可以换个角度，根据2009年W. Chen等人[23]的思路，可以预先通过一个随机函数判断节点能否通过边成功影响。如果则边边可以选择作为影响力传播的边，否则认为边不被选择作为影响力传播的边并移除该边，这样就可以得到一个新的图。对于一个节点，将可到达的节点计做，节点数量是，节点的影响力是。如果想要求取影响力增益，则需要先从中删除,及它们的边得到。如果节点加入S会带来影响增益计作，那么计算如公式（3.7）所示：

（3.7）

以图3-1为例，图中有4个节点，7条边，假设每条边的权值均是0.5，那么该图的边的权值矩阵如表3-2所示，第1行和第1列代表节点的编号。假设当前的种子节点只有2号节点，现在需要求节点0，1，3其中一个节点作为新种子节点会带来的影响力增益。

表3-2 独立级联模型中有向图1的权值矩阵

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** |
| **0** | 0 | 0.5 | 0 | 0.5 |
| **1** | 0 | 0 | 0.5 | 0.5 |
| **2** | 0 | 0 | 0 | 0.5 |
| **3** | 0 | 0.5 | 0 | 0 |

通过对每条存在的边，随机给一个0到1之间的随机数字，可得到随机数矩阵，如表3-3所示。

表3-3 随机数矩阵

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** |
| **0** | 0 | 0.3 | 0 | 0.35 |
| **1** | 0 | 0 | 0.55 | 0.25 |
| **2** | 0 | 0 | 0 | 0.15 |
| **3** | 0 | 0.15 | 0 | 0 |

如果对于，如果，则移除该边。最终可得到新图，如图3-2所示。

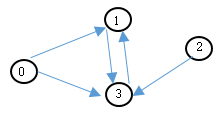


图3-2 随机级联有向图

先删除种子节点2及其可达的节点，可得到新图，如图3-3。



图3-3 影响增益图

根据图3-3，由公式（3.3）可求得节点的影响力增益如表3-4所示。

表3-4 独立级联模型节点的影响力增益

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **节点编号** | 0 | 1 | 2 | 3 |
| **影响力增益** | 1 | 0 | 0 | 0 |

### 3.3.3热量传播模型

为了捕捉市场临时的信息传播，2008年Hao Ma等人[20]提出了使用热量传播模型来对复杂的社交网络的影响力传播模型（Heat Diffusion Model, HDM）进行建模。HDM模型提供了很多可以模拟真实条件的参数，比如时间和传导率。在选择商品促销的时候，存在一个时间的概念，即购物者会考虑邻居商品的时间，所以该实验使用HDM模型来对商品网络影响力传播建模。

在给定的一个有向图,其中*V*是节点的集合，*E*是边的集合，*P*是边上的权值的集合。为了简化模型，需要先做出如下的假定：

（1）在时间，节点传播出去的温度记作。

（2）是均匀传播出去的，传播的温度按度数进行均分。

（3）所有的节点有相同的热量传播能力。

（4）热量的传播方向跟边的方向有关，且不受温度差影响。

根据Hao Ma等人[20]的文章，节点*i*在*t*+时间内的变化等于它接受邻居的温度减去节点*i*传播给邻居的温度,经过时间*i*节点的温度如(3.8)所示，该公式可化简为(3.9)，*H*称为影响力传播矩阵，其计算如公式（3.10）所示。

(3.8)

(3.9)

(3.10)

代表节点*j*存在的概率，实验中会取1。用于标识节点*i*的出度，*i*的出度大于0时候，则为1，否则为0，是节点*i*的出度。

当时候有(3.11)：

(3.11)

求积分后可得(3.12)：

(3.12)

以图3-1为例，权值矩阵如表3-2所示。根据公式（3.10）可求得影响力传播矩阵，如表3-5所示。

表3-5 影响力传播矩阵

|  |  |  |  |
| --- | --- | --- | --- |
| -0.5 | 0 | 0 | 0 |
| 0.25 | -0.5 | 0 | 0.5 |
| 0 | 0.25 | -0.5 | 0 |
| 0.25 | 0.25 | 0.5 | -0.5 |

假设2是种子节点，初始温度是10，其余节点的温度是0，所有节点的初始温度记作。设置时间*t*=0.1，这个时间表示与邻居节点影响的时间，传导率=1，现在想观察影响力传播的过程。使用公式3.12求得，然后将给，重复进行这两个步骤30次，可得到30t时间内所有节点的温度随时间的变化的情况，从图3-4所示，节点的温度传播过程，各个节点的温度会趋于稳定。

图3-4 热量传播过程

在购物者购物的时候，并不会考虑那么多层的邻居商品。在本文的实验中，认为购物者只会考虑购物篮中存在的商品(被激活的节点)的存在直接关联性的商品(第一层邻居节点)。求种子节点1的影响力的过程是，先假设多有节点的激活温度是0.24，给节点1初始温度10，然后使用公式3.10求得，每个节点的温度如表3-6所示：

表3-6 图1热量传播模型的影响力

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **节点0** | **节点1** | **节点2** | **节点3** |
| **温度** | 0 | 9.518 | 0.238 | 0.243 |

此时大于激活温度0.24的节点有节点1和节点3，所以只有两个节点被激活。此时节点1的影响力是2。

### 3.3.4其他传播模型

除了上述的三种模型，还有很多的影响力模型，这里只简单介绍两个其他的影响力模型。

（1）传染病模型(epidemic model)

传染病模型最初用作对病毒在人群中的传播进行建模，现在已经扩展到对信息传播研究上了。在该模型节点可能存在三种状态，易感S(susceptible)，感染I(infected)，治愈R(susceptible)。该模型根据状态根据节点状态的变化可区分为三种模型，SI模型，SIS模型，SIR模型。SI模型中节点可由易感变成感染状态并一直保持感染状态。SIS模型中允许节点在易感状态和感染状态来回变化。SIR模型模拟了节点的免疫过程，节点由易感状态可切换到感染状态，当节点从感染切换到治愈状态后便不会再被感染。

（2）选举模型(voter model)

选举模型的原本用在统计物理中，现在被扩展到社交网络影响力的传播上了。该模型影响过程是，一个节点的状态由的状态来决定，通过随机从中选取一个状态作为的当前状态。这种变化模拟了在选举过程中意见在个体间的传播，一个节点通过与其他节点的交流并接受了其中一个节点的观点。这个模型常用于了解在某一个时刻网络中节点的状态分布和相关性质。

## 3.4影响最大化关键技术

这一节主要讲述影响最大化求解算法概述和在选取影响力最大的商品集合中会用到的有向图快速层次聚类算法。

### 3.4.1影响最大化问题求解算法概述

影响最大化问题是一个组合最优化的问题，很难找到全局最优解，所以很多研究者基于启发式算法和贪心算法来求解影响最大化问题。

（1）基于贪心算法

贪心算法不从整体考虑问题的最优解，它的特点是通过不断求取当前的最优解来取得局部最优解。

2003年，Kempe等人[15]指出在独立级联模型和线性阈值模型下,影响最大化节点集合的选取是一个NP难问题，因此Kempe等人使用贪心算法BasicGreedy每次选择一个使影响增益最大的节点加入到种子集合*S*中，最终求得含K个节点的影响最大化种子集。在求解过程中，种子集合*S*的影响力增益是非负且单调递减的，即满足子模特性，又因，所以该方法所求的近似解至少是最优解的,其中e是常数。

2008年Hao Ma等人[[[24]](#endnote-24)]认为热量在传播媒介上随时间的扩散与影响力的传播相似，因此创新性的使用热量传播模型（Heat Diffuse Model， HDM）来模拟复杂的影响力传播。在该模型中，Hao Ma等人求第*t*轮种子集合时候，使用第*t*-1次求取的种子集*S*与每个一起求取影响力，然后能与*S*一起影响力最大的加入到种子集S中。

2007年Leskovec J等人[16]发现BasicGreedy贪心在独立级联模型和线性阈值模型求解影响最大化问题过程中，影响力增益呈现出很好的子模特性，该性质下选取影响增益最大的种子节点时候，可以不更新所有节点的影响力增益，因此在BasicGreedy基础上提出了给节点增加影响力更新标识的CELF算法。该算法的求解精度与BasicGreedy相同，而求解速度正常情况比BasicGreedy算法快700倍。

2013年，Hui Li等人[19]认为在社会活动中人们会表现出从众意识（Conformity Aware）。在对从众行为进行影响力传播建模后，Hui Li等人证明社区划分后，原始的贪心算法在从众意识影响力传播模型上同样满足子模特性，且贪心解依然是最优解的。所以Hui Li等人提出一种基于社区划分的贪心算法，该算法分为两个阶段第一阶段使用类似于CELF算法对每个社区进行求解影响力增益最大的节点，然后不断的从所有社区的影响力增益最大的节点作为种子节点，并只在被选取种子节点的社区中，更新节点影响力增益，重复步骤直到选取*k*节点的影响力种子集合。

（2）基于启发式算法

启发式算法利通过分析节点的特性，依据经验选取更符合种子节点特性的节点来构造种子集合。

2011年jiang等人[[[25]](#endnote-25)]使用模拟退火算法来求解影响最大化问题，首先设置一个初始温度*T*,临界温度并随机选择一个初始的*k*节点种子集合*S*，然后随机的从邻居节点中选择一个节点来替换S中的一个节点得到新的种子集合，如果影响增益，则让，否则再通过一个收敛的随机函数（与*T*有关）来决定是否仍用替换*S*作为当前的种子集合。重复该步骤并让，直到结束程序。该算法的时间复杂度是O(*IRM*)，其中*I*是迭代的次数，*M*是边数，*R*是计算影响力增益的时间复杂度。

2012年Wang C等人[[[26]](#endnote-26)]使用局部的影响力来近似作为节点在图中的影响力的PIMA算法。该算法首先通过dijkstra算法构建节点间最大影响力路径(maximization influence path, mip)，然后忽略概率小于阈值的mip,这样可以高效的得到节点的局部影响力。这种方法大大减小的求解影响力的时间花费，PIMA算法比当时经过优化的蒙特卡洛算法快1000倍。

2014年Chen等人[21]为了缩小影响力种子候选集合的大小以及避免影响力种子集中于一个社区，提出了基于热量传播模型的影响最大化方法。该方法根据社区的种子选取情况，节点的度数和在网络中的位置等特征来选取种子节点集合，方法分为三步骤，第一社区发现，第二选取候选集合，第三通过不断试探性的替换种子集合的节点优化种子集合。

### 3.4.2有向图快速层次聚类算法

本文求影响最大化商品集合过程中会用到有向图的快速层次聚类，原因有两个，第一社区划分可以防止选取的种子集合中的商品集中于一个社区，第二可社区的划分能够反映哪些商品的类别存在较强的影响关系。

在本文使用的聚类算法是有向图的快速层次聚类，因为聚类不是本文的重点，所以只介绍在算法实现过程中用到的公式(3.18)和(3.20)，以及介绍算法的流程。

给定一个有向图，节点的带权出度用表示,带权入度则用来表示，根据2008年E. A. Leicht1等人[[[27]](#endnote-27)]的文献可知有向图的模块值计算公式为(3.10)。

(3.10)

(3.11)

(3.12)

(3.13)

当节点和节点在同一个社区的时候，为1否则为0。

引入(3.14)，其中*x*和*y*是社区的编号。当x=y的时候。

(3.14)

社区x和社区y连接的所有边的边权占图总的边权和比例用表示，则有：

(3.15)

社区x中节点入度和占图总的边权和比例用为表示，则有：

(3.16)

社区x中节点的出度和占图总的边权和比例用表示，则有：

(3.17)

有向图的模块值计算(3.10)可根据(3.14)～(3.17)简化为：

(3.18)

由有向图公式模块值计算公式(3.18)可得一个孤立点加入社区后的模块值的增益的计算公式(3.19)：

(3.19)

(3.19)可简化为(3.20)

(3.20)

表示当前社区*C*内的边权和，

是节点*x*连接*C*内部节点的边的权值和，

是终点是社区*C*的节点的边的权值和，

是边的起点是社区*C*的节点的边的权值和。

基于Blondel等人[[[28]](#endnote-28)]的无向图快速层次聚类算法的思想和E. A. Leicht1等人[27]计算有向图模块值的思想，设计了有向图的快速层次聚类算法（Fast Hirearchy Clustering, FHC）。

*FHC*算法的伪代码如表3-2所示，输入有向图*G*和误差阈值*ε*=0.001，输出社区划分后得到的社区列表*subg*，*subg*[i]是i社区。第1行至第2行是将每个节点加入*i*社区的节点集合中，即将每个节点看作是一个社区。第3行是在当前社区划分的情况下，根据公式(3.18)计算整个网络的模块值。第5行至16行将每次选择一个节点，将其看作是一个社区，然后计算将该节点加入到邻居社区带来的模块值增益*irc*，在这个过程中记录最大的模块值增益*max\_irc*及其邻居社区编号*best\_cid*，然后将节点加入到编号为*cluster*[*best\_cid*]的社区中。第4行至19行中*Q*上次调整所有节点所属社区的整个网络的模块值，是这次调整所有节点所属的社区后的整个网络的模块值，这段代码是作用是调整节点所属社区直至不满足条件。第20行是新建一个空列表*subg*用于存放社区。第21行至24行是根据cluster中每个社区的节点集合从原图*G*中抽取出子图*new\_g*，然后将子图*new\_g*加入到图列表*subg*中。第25行返回子图列表*subg*。

表3-2 *FHC*算法伪代码

|  |
| --- |
| Algorithm :*FHC*  input：*G*,=0.001  output：*subg*   1. for each *i* in *len*(*G.V*)   2. *cluster*[*i*] #将每个节点看作是一个社区  3.*cluster*划分状态下，用公式(3.18)计算模块Q值  4.do  5. for each in *G.V*  6. *max\_irc* -9999  7. *best\_cid* = 0  8. *i\_id*所属社区的编号  9. remove from *cluster*[*i\_id*]  10. *nei\_cluster*的邻居社区编号  11. for each *cid* in *nei\_cluster*  12. *irc*用公式(3.20)计算节点加入*cid*社区后模块值变化值  13. if *ircmax\_irc*  14. *max\_ircirc*  15. *best\_cidcid*  16. *cluster*[*best\_cid*]*.append*()  17. *Q*  18. 在*cluster*划分状态下，用公式(3.20)计算模块*Q*值  19.while - Q >  20.*subg empty list*  21.for each *v\_set* in *cluster*  22. if *len*(*v\_set*)>0  23. *new\_g*根据*v\_set*节点集合从*G*中抽取子图  24. *subg.append*(*new\_g*)   1. 25.return *subg* |

## 3.5本章小结

本章先介绍了社交网络的定义，以及给出阐述社交网络理论会用的一些符号及其含义。在这个基础介绍社交网络影响最大化化问题的定义，然后介绍线性阈值模型，独立级联模型，热量传播模型等传播模型。最后介绍求解影响最大化问题的关键技术，包括求解影响最大化问题的算法概述和本文在购物篮分析中会用到的快速层次聚类算法及其实现。

# 第四章 基于商品影响力的购物篮分析

## 4.1引言

本章设计了基于商品影响力的购物篮分析算法，用于求解促销商品，这个算法包含了三个重要的步骤，第一步是构建商品网络。商品网络的边权值表示商品间的相关性强度，基于apriori算法思想，编写求取两两商品间相关性的伪代码。第二步是对商品影响传播进行建模，分别基于独立级联模型和热量传播模型的思想设计了相应的求解商品影响增益的算法。第三步是设计一个基于社区划分的影响最大化算法，该算法的框架源于Hui Li等人[19]的基于社区划分的影响最大化算法，不同之处是本文所以用的社区划分算法是有向图快速层次聚类[11][12]，而不是cluto软件包的*l-way partition*算法[[[29]](#endnote-29)]，为了让该框架同时适用于上述两种影响传播模型省去了优化效率的子模函数，并且在求解影响增益最大的商品的过程中加入了社区计分，如果商品具有相同的影响增益，则选择社区计分大的商品，即影响潜力更大的商品。

## 4.2商品网络的构建

在超市的中，顾客往往会把想买的东西一起放在购物篮中，收银员通过扫描条形码将购物篮中的商品录入到电脑中，一次购买的商品有一个共同的交易号。因此同一交易号的商品组合在一定程度上反映了顾客喜欢同时购买这些商品，使用apriori关联规则算法我们可以得到顾客购买了一种商品购买另外一种商品的可能性，即顾客购买一个商品后，该商品会激起该购买另一件商品的欲望。我们将apriori算法求得的这种商品关联性当作是商品之间的影响关系，关联性越强，商品间相互影响的关系也越强。

实验中我们将使用一个有向带权网络*G*(*V*,*E*,*W*)来表示商品的影响关系网络，其中表示商品网络中的一件商品，表示商品节点会影响到商品节点，商品的影响关系强度是，可根据apriori算法的思想求得。

根据2.3节apriori算法的思想，求两两商品间的影响强度的由5部分组成，一生成1项集，二根据*k*项集生成*k*+1项集，三求项集的支持度，四根据项集的支持度求两两商品间的关联性。为方便理解伪代码，先给出一些变量的含义如表(4-1)。

表4-1 独立级联模型算法伪代码中的一些变量的意义

|  |  |
| --- | --- |
| *Dataset* | *dataSet*是一个二维列表，每一行代表一次交易的商品组合 |
|  | 是项集的列表，里面的每个项集的元素个数是相同的 |
|  | 2项集列表 |
| *minConfident* | 最小置信度 |
| *minSupport* | 最小支持度 |
| *supportData* | 每个项集的支持度 |

（1）构造1项集的算法是*createL1*。

*createL1*算法的伪代码如表4-2输入*dataSet*，一个二维列表，每一行代表一次交易的商品组合,最后输出所有1项集列表。第1行是创建一个列表用于存放1项集，第2行到第5行是从*dataset*中获取每个商品组合*transaction*，如果*transaction*中的只含有商品*item*的列表不在1项集列表中，则将该只含*item*的列表加入到中。第6行返回1项集列表。

表4-2 *createL1*算法伪代码

|  |
| --- |
| Algorithm: *createL1*  Input: *dataset*  output:  1.←  2.for each *transaction* in *dataset*  3. for each *item* in *transaction*  4. if [*item*]not in  5. insert [*item*] to  6.return |

（2）通过*k*项集构造*k*+1项集的算法是*genLn*。

*genLn*算法的伪代码如表4-3所示，输入*k*项集列表，输出*k*+1项集列表*。*第1行是创建一个列表*Ln*用于存放*k*+1项集。第2行是获得列表的长度，存放至*lenLk*中。第3行是获取中存放的*k*项集中的元素个数加一后存放至*n*中，这里*n*等于*k*+1。第6行和第7行分别是是对第*i*个*k*项集中的元素和第*j*个*k*项集中的元素升序排列，分别得到和。第8行至第12行是判断和中前*n-2*个元素是否相同，如果相同则合并和项集得到*k*+1项集，并将新项集加入到中。第4行至第12行是根据对项集裂变的*k*项集两两组合尝试构造出*k*+1项集，并将*k+1*项集加入到中。第13行返回*k*+1项集列表。

表4-3 *genLn*算法伪代码

|  |
| --- |
| Algorithm: *genLn*  input:  output：  1.←  2.*lenLk*←*len*() # 项集列表的长度  3.*n←len*([0])+ 1 # 是*k*项集列表，项的元素个数加1赋值给*n*  4.for each *i* in [0,1...*lenLk*-1]  5. for each *j* in [*i*+1,*i*+2...*lenLk*-1]  6. ← *sorted*( *list*([*i*]) ) # 第*i*个项集的元素升序排列  7. ← *sorted*( *list*([*j*]) ) # 第*j*个项集的元素升序排列  8. *C1* ←[*:n*-2] # 将的前*n*-2个元素赋值给*C1*  9. *C2* ←[*:n*-2] # 将的前*n*-2个元素赋值给*C1* |

续表4-3 *genLn*算法伪代码

|  |
| --- |
| 10. if *C1* equal to *C2*  11. if [*i*]∪[*j*]not in  12. add [*i*]∪[*j*] to  13.return Ln |

（3）计算项集的出现频率的算法是*scanD*。

算法*scanD*的伪代码如表4-4，输入*dataSet、、minSupport*，其中*dataSet*是一个二维列表，每一行代表一次交易的商品组合，是*k*项集的列表，*minSupport*是最小支持度*。*第1行是新建一个空字典*ItemsetCnt*，字典的键是*k*项集，值是*k*项集的出现次数。第2行至第8行是计算*k*项集列表中的*k*项集*itemset*在数据的集合*dataSet*中出现的次数，并存放到*ItemsetCnt*中。第9行是获取数据集*dataSet*的记录数量存放至*numItems*。第10行新建一个空列表*ret\_*，用于存放频繁项集。第11行新建一个空的字典*supportData*，字典的键存放频繁项集，值是频繁项集的支持度。第12行至第15行是计算*k*项集列表中的*k*项集的支持度，如果项集支持度不小于最小支持度则认为该项集是频繁项集，并将该频繁项集加入到*ret\_*中，然后将该频繁项集作为键，支持度作为值加入到字典*supportData*中。第17行返回频繁项集和频繁项集的支持度。

表4-4 scanD算法伪代码

|  |
| --- |
| Algorithm: *scanD*  input： *dataSet*，，*minSupport*  output：*ret\_*，*supportData*  1.*ItemsetCnt*←  2.for each *tid* in *dataset*  3. for each *itemset* in  4. if *itemset* is not a subset of *tid*  5. if *itemset* is not one of key in *ItemsetCnt*  6. *ItemsetCnt*[ *itemset* ]←1  7. else  8. *ItemsetCnt*[ *itemset* ]*←ItemsetCnt*[ *itemset* ]+ 1  9.*numItems*←float(length of dataSet)  10.*ret\_*←  11.*supportData*←empty dictionary  12.for each *key* in *ItemsetCnt*  13. *support*←*ItemsetCnt*[*key*] /*numItems*  14. if *support* >= *minSupport*  15. insert *key* to *ret\_*  16. *supportData*[*key*]←*support*  17.return *ret\_*，*supportData* |

（4）计算任意两个商品小类间的关联规则的算法是*generateRules*。

算法*generateRules*的伪代码如表4-5所示，输入*，supportData*，*minConfident*。其中是二维项集列表，是2项集列表，*supportData*是字典，键是项集，值是项集的支持度，*minConfident*是最小置信度。第1行是新建空的列表*ruleList*用于存放规则，每个元素是一个包含项集1，项集2和置信度的三元组，表示项集1项集2的置信度。第2行获得项集列表长度存放至*len\_*中。第3行至第9行对2项集列表*L2*中的2项集，计算一个项集与另一个项集之间的置信度，如果置信度不小于最小置信度，则将该规则加入到*ruleList*中。第10行返回规则列表*ruleList*。

表4-5 generateRules算法伪代码

|  |
| --- |
| Algorithm: *generateRules*  input：*，supportData*，*minConfident*  output：*ruleList*  1.*ruleList*←  2.*len\_*←length of  3.for each *itemset* in  4. for *item* in *itemset*  5. if *item* not in *supportData.keys* or not in *supportData.keys*  6. continue  7. *confidence*←*supportData*[*item*]/*supportData*[*itemset*{*item*}]  8. if *confidence* >= *minConfident*  9. insert (*itemset*{*item*}, *item*, *confidence*) in *ruleList*  10.return *ruleList* |

## 4.3基于独立级联模型的购物篮分析

独立级联模型是社交网络影响力传播模型的一种经典模型，该模型的特点是认为个体间的影响是相互独立，且只使用*i*-1轮激活的节点尝试一次激活*i*轮未被激活的节点。每个节点被激活的可能性可用(3.4)来计算。

通过分析购物者的购物行为，可发现在逛超市的过程中，购物者见到一件商品并加入购物篮中，刚放入购物篮中的商品会让购物者有可能性*p*想去购买其他相关的商品（该商品在商品网络中的邻居商品)，而apriori关联规则算法求得的商品关联性则反映了购买欲*p*。消费者购买一件商品后考虑相关商品的过程与独立级联模型的用刚激活的节点尝试激活新节点类似。商品的影响力可以定义为，一件商品能够影响购物购买其他商品的能力，在独立级联模型中使用激活的商品数量来表示。在进行商品促销决策时候，往往会想选择何种商品进行促销能够带来更多的销量。本文基于这样的应用场景采用基于独立级联模型的影响最大化算法尝试寻找出影响力最大的商品组合，为零售商家选择促销商品提供信息参考。

根据3.3.2节所述W. Chen等人[23] 模拟独立级联模型的思想，实现了求商品影响力增益的算法，将算法名字记为为*ICM*。

*ICM*算法的伪代码如表4-6所示，算法输入两个参数，有向图*G*和种子节点集合*S*，最后输出节点的影响力增益*f*。第1行是初始化影响力传播模拟次数*N*为2000。第2行新建一个所有元素值为0的数组*f*。第3行创建迭代计数变量*r*，初值为0。第4行至地15行模拟影响力传播*N*次，并统计每个节点如果加入种子集合会带来的影响力增益。其中第5行是复制一份图*G*为。第6行至地8行是随机的从图中移除边。第9行是从图移除种子节点的节点及其边，得到的新图赋值给。第10行是新建一个值为0的数组，用于统计每个节点影响力收益。第11行至第12行是计算每个节点可达的节点数量，包括自身，作为节点影响增益，存储到中。第13行至第14行统计每次模拟的影响力增益到*f*中。第16行至第17行求每个节点的影响力增益的平均值。第18行返回每个节点的平均影响增益*f*。

表4-6 ICM算法伪代码

|  |
| --- |
| Algorithm ICM(G)  Input: *G*，*S*  output: *f*  1.*N* ←2000 # 模拟影响力传播次数  2.*f←zeros array* # *f*初始是一个0数组  3. *r* ←0  4.*while r < N*  5. ←the copy of *G*  6. for each in  7. if  8. remove from  9. ←Remove node in S from  10. node\_gain = zeros array  11. for each in  12．  13. for each in *G.V*  14.  15. *r* ←*r* +1  16.for in *f*  17.  18.return *f* |

## 4.4基于热量传播模型的购物篮分析

热量在介质上的复杂传播过程，高温物体与低温物体接触过程中，高温物体的温度会逐渐减低，而低温物体的温度会逐渐的升高，并且接触时间的增加温度会渐渐趋于相同。为了对社交网络影响力传播建模，研究者对热量传播进行了简化和调整。在社交网络的热量传播模型下，每个个体*v*会按相同的比例的将影响平均传播给他的邻居，个体*v*的邻居越多那么每个邻居受到个体*v*影响就越弱，同时个体*v*受到邻居的影响是可以累积的。在不同时刻，活跃的节点数也有所不同，且随着影响时间的增加，每个个体的温度会趋于一个稳定值。2008年Hao Ma等人[20]使用公式(3.9)来计算带权有向网络中节点的被影响程度。

热量传播模型与独立级联的不同之处是，热量传播模型不仅仅是使用上次刚激活的节点来激活新节点，它考虑的是所有节点间的影响力在彼此之间的传播，一个节点的活跃程度可用温度来衡量，活跃的节点会让其他节点也变得活跃，而其他商品活跃度提高的同时自身的活跃度可能会有一定的降低。这可以用来模拟购物者在看到一些促销品的时候，突然想购买便顺手将其放入购物篮子中，然后又想购买相关的一些商品，在商品对比过程中，对相关商品的购买欲望可能会增加或降低，然后又把想购买的商品放入购物篮中。促销过程中促销品可以看做是热源，热量传播时间可看做是购物者购买促销品后考虑和比对其他商品的时间，最终购买的商品看做是这个促销品带来的影响。

（1）基于3.3.3所述2008年Hao Ma等人[20]的热量传播模型的思想，实现了该模型下求商品影响力的算法，将算法名字记为为*HeatDiffusion*。

算法*HeatDiffusion*的伪代码如表4-7所示，算法输入*G, S, h0, a, t,* ，其中*G*是有向图, *S*是种子节点集合, *h0*是种子节点的初始温度, *a*是热量传导率, *t*是与邻居节点的传播时间, 是节点的激活阈值，该算法最后输出种子集合的影响力。第1行创建一个变量用于存放影响力，初值为0。第2行创建一个数组*f*用于存放节点被影响的程度，初始化为0数组。第3行至第4行是初始化种子节点的温度为*h0*。第5行是根据公式3.9求取影响力传播矩阵*H*。第6行是计算每个节点的被影响程度被存储到*f*中。第7行至第9行计算被激活的节点数，并存放至中。第10行返回。

表4-7 *HeatDiffusion*算法的伪代码

|  |
| --- |
| Algorithm: *HeatDiffusion*  input：*G, S, h0, a, t,*  output：  1.←0 # 初始化影响力为0  2.*f*←*zeros array* # *f*初始是一个0数组  3.for each in *S*  4. ←*h0* # 初始化种子节点的温度为*h0*  5.根据式（3.9）计算*H*  6.  7.for each in *G.V*  8. if  9. += 1  10.return |

（2）本文设计了在热量传播模型中求节点影响力增益算法，名字记为*HeatDiffusionGain*。

算法*HeatDiffusionGain*的伪代码如表4-9所示，算法输入*G, S, h0, a, t,* ，其中*G*是有向图, *S*是种子节点集合, *h0*是种子节点的初始温度, *a*是热量传导率, *t*是与邻居节点的传播时间, 是节点的激活阈值，该算法最后输出每个节点加入种子集合带来的影响力增益。第1计算当前的种子集合的影响力存储到新建的变量*cur\_incluence*中。第二行是获得图G的节点数。第3行新建一个数组*f*，*f*所有元素初值为0。第4行至第9行尝试计算的影响力存放到新建的变量*influence中*，然后减去种子集合S的影响力*cur\_incluence*得到节点加入种子集合会带来的影响力增益。第10行返回每个节点如果作为种子节点会来的影响力增益*f*。

表4-8 *HeatDiffusionGain*算法伪代码

|  |
| --- |
| Algorithm: *HeatDiffusionGain*(*G, S, h0, a, t,* ):  input：*G，S， h0， a，t,*  output：  1.*cur\_incluence←HeatDiffusion*(*G, S, h0, a, t,* )  2.*num = len*(*G.V*)  3.*f←zeros*(*num*) # 0 数组，元素个数是*num*个  4.for each in *G.V*  5. if *node* in *S*  6. continue  7. *new\_seed\_set =*  8. *influence = HeatDiffusion*(*g, new\_seed\_set, h0, a, t,* )  9. = cur\_incluence – influence  10.return *f* |

## 4.5基于社区划分的影响最大化算法

本文设计一个基于社区划分的影响最大化算法，该算法的框架源于Hui Li等人[19]的基于社区划分的影响最大化算法，不同之处是本文所以用的社区划分算法是有向图快速层次聚类[11][12]，而不是cluto软件包的*l-way partition*算法[[[30]](#endnote-30)]，为了让该框架同时适用于上述两种影响传播模型省去了优化效率的子模函数，并且在求解影响增益最大的商品的过程中加入了社区计分，如果商品具有相同的影响增益，则选择社区计分大的商品，即影响潜力更大的商品。

社区计分（*score*）的计算见式（4.1）。其中*num\_node*是社区的节点个数，*seed\_num*是当前选取的种子数量。社区计分基于这样的一个概念，社区的每个种子节点影响的节点数量是相同的，且社区中的种子节点会影响所有的节点，如果社区选择的种子数越多那么平均每个节点影响的节点数就会变少，即社区的计分会变小。如果两个节点的影响增益相同，则选择社区计分大的种子节点，因为该种子节点会比社区计分小的种子节点在实际中潜在的影响力可能会更大。

(4.1)

本文设计的基于社区划分的影响最大化算法的流程见图4-1。

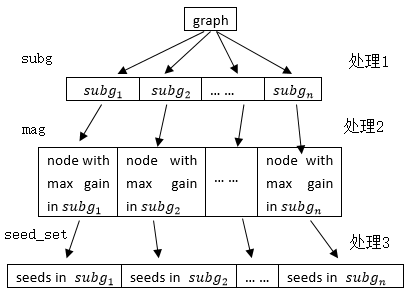


图4-1 基于社区划分的影响最大化算法流程

处理1是指将图划分成多个子图，记作*subg*。第*i*个子图用表示。处理2中的第i个位置存放的影响增益最大的节点及其影响增益，当*mag*的第*i*个位置的节点被选为种子节点，从子图中寻找影响增益最大的节点更新*mag*列表第*i*个位置存放的信息。处理3是指从所有社区中选择影响力增益最大的节点，如果节点影响增益相等，则选取所在的社区*score*值最大的那个。

假设有一个7个节点11条边的有向带权商品网络如图4-2所示。以该商品网络为例子，寻找两个促销商品，这两个商品能尽可能的带动商品的销售。



图4-2 商品网络

假设图4-2的商品网络经过社区划分的结果如图4-3所示。



图4-3 社区划分结果

假设每个社区的影响增益最大的节点如表4-9所示。此时节点1和节点2拥有相同的影响增益，但节点1所属社区的社区计分更高，所以选择1号节点为种子节点。

表4-9 社区中影响增益最大节点

|  |  |  |  |
| --- | --- | --- | --- |
| **社区编号** | **节点编号** | **最大影响增益** | **社区计分** |
| 1 | 1 | 2 | 4 |
| 2 | 4 | 2 | 3 |

重新从节点1所属的社区选择新的影响增益节点，可得到表4-10。此时4号节点影响增益最大，所以选择4号节点作为种子节点。此时已经选取了两个种子节点，即两个促销商品，流程结束。

表4-10 社区中新的影响增益最大节点

|  |  |  |  |
| --- | --- | --- | --- |
| **社区编号** | **节点编号** | **最大影响增益** | **社区计分** |
| 1 | 2 | 1 | 2 |
| 2 | 4 | 2 | 3 |

为方便理解伪代码，先给出部分变量及其含义，如表4-10所示。

表4-10 贪心算法伪代码重要变量的含义

|  |  |
| --- | --- |
| *k* | 种子节点数 |
| *seed\_set* | 种子集合 |
| *dfm\_gain* | *HeatDiffusionGain*或*ICM*算法 |
| *Mag* | 是一个列表，*mag*[*i*][0]存放*i*社区当前影响增益最大的节点, *mag*[*i*][1]存放该节点的的增益 |
| *Subg* | 是一个社区列表，*subg*[*i*]的当前影响增益存放在*mag*[*i*][*0*] |
| *FHC* | 是3.4.2节介绍的有向图社区划分算法 |

基于社区划分的影响最大化算法是*get\_seed\_set*，算法伪代码如表4-11所示。这个算法还包含三个其他算法，分别是*FHC*算法*、mag*算法*、arg\_max\_mag*算法和*dfm\_gain*算法。其中*FHC*算法是有向图快速层次聚类算法。*init\_mag*算法是初始化社区影响增益最大节点列表的算法。*arg\_max\_mag*算法是获取*mag*中影响增益最大节点在*mag*中的索引编号的算法。

*get\_seed\_set*算法输入的参数有两个，有向图*G*和种子个数 *k*。最后输出影响力最大的*k*节点种子集合。第1行是使用有向图层次聚类算法对图*G*进行社区划分得到社区列表*subg*。第2行至第6行是初始化影响增益最大节点列表*mag*、种子集合*seed\_set*和记录循环次数的变量*i*。第7行至第12行每次从*mag*列表中选取一个影响力增益最大的节点加入到相应社区的种子集合中，并从相应社区寻找新的影响力增益最大的节点更新*mag*，其中第11行是计算独立级联模型中所有节点的影响力增益。第14行是获得影响力增益最大的节点及其影响增益，并存放至*mag*列表的第*idx*个位置。第15行是返回影响力最大的种子集合。

表4-11 *get\_seed\_set*算法伪代码

|  |
| --- |
| algorithm: *get\_seed\_set*  input: *G*, *k*  output: *seed\_set*  1. *subgFHC*(*G*)  2. *maginit\_mag*(*subg*)  3. *seed\_set*  4. for each *i* in[0,1,2…,*len*(*subg*)]  5. *seed\_set.append*([])  6.*i*0  7. while *i* < *k*  8. *idxarg\_max\_mag*(*mag, seed\_set, subg*)  9. s*eed\_set*[*idx*][0]*.append*(*mag*[*idx*][0])  10. *seed\_set*[*idx*][1]*seed\_set*[*idx*][1] *+mag*[*idx*][1]  11. *subgthsubg*[*idx*]  12. *f = dfm\_gain* (*subgth, seed\_set*[*idx*][0])  13. *mag*[*idx*] *select the node with max gain from f*  14. *ii+*1  15.return *seed\_set* |

*init\_mag*算法用于初始化*mag*列表，该算法的伪代码如表4-12。输入子图列表*subg，*最后输出*mag*列表*。*第1行新建一个空列表*mag*。第2行至第3行对于每个子图在*mag*中添加一个包含节点编号和最大影响增益的列表，*mag*[*i*]是第*i*个列表，第一个元素是社区*i*的影响增益最大的节点，第二个元素是该节点的影响增益。第5行返回*mag*列表。

表4-12 *init\_mag*算法伪代码

|  |
| --- |
| Algorithm: *init\_mag*  input: *subg*  output: *mag*  1. *mag*  2. for each *subgth* in *subg*  3. *node, gainget\_node\_with\_max\_gain*(*subgth,* [])  4. *mag.append*([*node, gain*])  5.return *mag* |

*arg\_max\_mag*算法是获取*mag*中影响增益最大的节点在*mag*的索引编号，该算法的伪代码如表4-13。算法输入存放每个社区影响增益最大的节点及其增益的*mag*，当前的种子集合*seed\_set*，社区列表*subg，*最后输出*mag*中影响力增益最大的节点所在的索引*。*第1行是创建一个新的存放影响增益最大节点的列表*new\_mag*，该列表将会存放多个3元组，每个3元组的内容包含节点在*mag*的位置，取整后的种子节点影响增益以及社区的*score*值。第2行至第8行对于*mag*的第*idx*个元素，根据式(4.1)计算相应的社区计分，然后将*idx*，*mag*[*idx*]存放的影响力增益和社区的*score*值存入到*new\_mag*中。其中第7行是将浮点型的影响增益强制转换为整型。第9行至第10行是优先选择*new\_mag*中影响增益最大的那个三元组，影响增益相同则选择社区打分最大的那个三元组。其中*key*是一个匿名函数，该函数获取输入的三元组的最后两个元素的内容，即影响增益和社区打分。第11行返回索引编号*idx*。

表4-13 arg\_max\_mag算法

|  |
| --- |
| Algorithm: *arg\_max\_mag*  input: *mag, seed\_set, subg*  output: *idx*  1.*new\_mag*  2.for each *idx* in [0,1,2,…, *len*(*mag*)]  3. *nodemag*[*idx*]  4. *numsubg*[*idx*]*.number\_of\_node*  6. *score = num /* (*len(seed\_set*[*idx*])+1)  7. *influ = int*(*node* [0]+0.3)  8. *new\_mag.append*([*idx, influ, score*])  9. *key=lambda x :* [*x*[1]*,x*[2]]  10.*idx, influ, scoremax*(*new\_mag, key*)  11.return *idx* |

## 4.6本章小结

本章主要设计了基于商品影响力的购物分析算法，用于解决促销商品的选择问题。首先对商品网络的构建进行伪代码设计，因为构建商品网络只需求商品两两之间的关联性，所以简化了apriori算法，设计了求出1项集和2项集的支持度，然后求出商品间的关联的算法。本章还分析了独立级联模型和热量传播模型与商品影响传播的相似性及实现了基于这两个模型的求解商品影响增益的算法，最后设计了基于社区划分的影响最大化算法，用于求解能使影响最大化的*k*件商品，并解决选择促销商品的问题。

# 第五章 实验

## 5.1引言

该章主要探究上一章设计的基于独立模型和基于热量传播模的影响最大化算法在实际数据集中的表现。在该章会介绍实验的背景，实验数据集，实验结果评估方式，然后展示和分析实验结果。

## 5.2实验背景

### 5.2.1实验环境

本文所有实验均是在一台计算机上完成，计算机主要配置如下：

处理器：Intel(R) Celeron(R) CPU 1000m @ 1.80GHz 1.80GHz

安装内存：4.00 GB(3.85 GB 可用)

硬盘内存：1000G；

操作系统：Windows 8.1，64位系统，基于x64的处理器

实验所用的算法均采用python语言在ipython notebook工具下实现。

### 5.2.2实验数据集

本文使用的数据集来自一个真实的超市pos机记录的交易记录，取了其中一个月作为本实验的数据集。该数据集含有的字段信息如表5-1所示。

表5-1 实际数据集的含有的字段

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 部门编号 | 销售数量 | 经营小类 |
| 销售流水号 | 商品小类 | 销售额 | 经营小类名称 |
| 商品名称 | 商品小类名称 | 销售时间 | 品牌 |
| 项次 | 商品中类 | 品牌名称 | 毛利 |
| 商品ID | 商品中类名称 | 商品种数 | 会员编号 |

构建商品网络前，首先要构建商品的组合数据。从原始数据集合筛选出销售流水号和商品小类这两列集，同一销售流水号码标识商品小类在统一购物篮中。然后根据销售流水号筛选出每次购买的商品组合。部分商品组合如表5-2所示所示，每一行代表一次交易，一次交易的购买的商品用空格隔开。

表5-2 商品组合

|  |
| --- |
| 礼盒 橄榄菜 |
| 成人牙膏 洗护发套装 桶装面 益生菌乳酸奶 专柜其它面包 超市塑料袋 |
| 散装糕点 含片类糖果 |
| 其它保健品 |
| 散装糖果 纯牛奶 散装巧克力 |
| 其它餐具 散装糕点 成人牙膏 |

续表5-2 商品组合

|  |
| --- |
| 酸奶 超市购物袋/篮 其它零食 |
| 其它茶饮料 海产制品 洗衣粉 含乳饮料 散装糖果 橙类 |
| 花果类 超市塑料袋 进口糖果 叶菜类 其它浆果 |
| 其它餐具 成人牙膏 |
| 酸奶 超市购物袋/篮 |
| 洗衣粉 含乳饮料 散装糖果 |
| 礼盒 橄榄菜 |
| 成人牙膏 洗护发套装 桶装面 |
| 散装糕点 含片类糖果 |
| 益生菌乳酸奶 专柜其它面包 超市塑料袋 |

使用4.2节的*generateRules*算法构建商品网络。实验构建网络选择的最小置信度1/179164，支持度均是0.001。以用节点表示商品，边表示商品间的关联性，边的权值表示关联性的强度，构建的网络有1039个节点，282192条边。

### 5.2.3商品特征的选择

选出的影响最大商品组合，因为没有实际促销活动的结果供参考和验证，所以在本文使用商品的基本特征和在网络中表现的特征来描述实验结果并分析实验结果的合理性。本文选取了四个特征来描述影响最大化商品的特点，第一个是反映销售量情况的特征（销售量排名），其余三个特征分别是带权出度排名、带袋权入度排名和*pagerank*值排名。

1、商品销售量排名。用来表示影响最大化商品的销量与畅销商品的关系

2、带权出度排名。用来表示影响最大化商品直接影响邻居节点的强度。

3、带权入度排名。用来表示影响最大化商品受邻居节点直接影响的强度。

4、*pagerank*值排名。在2014年戚戚[13]认为商品间的联系与网页间的联系相似，为了挖掘出一些重要的商品，即潜在的可能带动销售的商品，使用*pagerank*算法求出商品的*pagerank*值，排名靠前的则认为更可能是重要的商品。本文将使用该方法查看影响最大化商品的*pagerank*值的排名情况。*pagerank*的两个参数阻尼系数（gehpi软件中称为概率*p*），收敛阈值（gehpi软件中称为误差*epsilon*），分别设置为0.85和0.0001。

根据商品网络的特征，以商品销量排名为横坐标，以带权值出度排名为纵坐标绘制散点图如图5-1所示。以商品销量排名为横坐标，以pagerank值排名为纵坐标绘制散点图如图5-2所示。从图可以看出带权出度排名与销量排名并没有明显的相关性，而pagerank排名与销量排名呈现正相关性，所以在分析商品特征时，只用销量排名、带权出度排名和带权入度排名分析商品的特征。

图5-1 带权出度排名与销量排名的关系图

图5-2 pagerank排名与销量排名的关系图

## 5.3实验结果

### 5.3.1社区划分结果

网络被划分成7个社区，特征如表5-3所示。

表5-3 社区特征

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **社区** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 节点数 | 258 | 150 | 176 | 208 | 235 | 8 | 4 |
| 边数 | 42090 | 913 | 9613 | 14389 | 25964 | 20 | 12 |
| 平均入度 | 163.1395 | 6.0867 | 54.6193 | 69.1779 | 110.4851 | 2.5000 | 3.0000 |
| 平均出度 | 163.1395 | 6.0867 | 54.6193 | 69.1779 | 110.4851 | 2.5000 | 3.0000 |
| 密度 | 0.6347 | 0.0408 | 0.3121 | 0.3341 | 0.4721 | 0.3571 | 1.0 |

每个社区中的商品小类所属的经营小类如表5-4。

表5-4 社区中的经营小类

|  |  |  |
| --- | --- | --- |
| **社区** | **数量** | **经营小类的名称** |
| 1 | 32 | 饼干糕点 禽蛋类 厨房用品 快餐 手机通讯 蔬菜 母婴用品 即食食品 畜肉类 消杀用品 水果 腌菜 日用小件 粮油 休闲食品 酒 水产 综合生鲜 奶制品 豆制品 休闲小吃 女性护理用品 中式面制品 饮料 家居清洁护理 副食品 应节食品 清洁工具 禽肉类 冲饮品 熟肉制品 冷冻冷藏食品 |
| 2 | 34 | 运动服装 皮具箱包 内衣 数码商品 彩电 家居用品 手机通讯 配饰 童装 冲饮品 婴幼产品 袜子 婴幼食品 生活电器 日用小件 休闲装 体育器材 男士服装 少淑女装 黄金珠宝 厨卫 碟机音响 办公用品 化妆品 儿童用品 洗衣机 功能服 厨房用品 (停用)生活配套 钟表 皮鞋 保健滋补品/药品 淑女装 冰箱 |
| 3 | 17 | 生活用纸 母婴用品 文化用品 数码商品 家居用品 消杀用品 家纺用品 汽车用品 眼镜 家居清洁护理 生活电器 厨房用品 日用小件 冲饮品 文化音像制品 清洁工具 床上用品 |
| 4 | 29 | 家纺用品 生活电器 面部护理用品 厨房用品 消杀用品 美容工具 母婴用品 成人用品 婴幼产品 婴幼食品 床用品 日用小件 生活用纸 宠物食品用品 女性护理用品 洗浴用品 男士洗护用品 洗发美发用品 床上用品 保健滋补品/药品 化妆品 口腔清洁用品 粮油 家居清洁护理 衣物清洁用品 身体护理用品 冲饮品 文化音像制品 糖果巧克力 |
| 5 | 27 | 饼干糕点 生活电器 面部护理用品 厨房用品 消杀用品 烘焙面制品 母婴用品 即食食品 饮料 婴幼食品 禽蛋类 日用小件 应节食品 酒 奶制品 男士洗护用品 文化音像制品 家纺用品 休闲食品 快餐 粮油 副食品 水产 冲饮品 保健滋补品/药品 糖果巧克力 冷冻冷藏食品 |
| 6 | 1 | 酒 |
| 7 | 1 | 空调 |

根据社区划分结果可以直观的估计商品小类间影响关系（关联性），以社区3为例，可以从商品小类图中抽取出经营小类间的关系，如图5-1所示。从这个图可以直观的看出各个商品小类之间的关系，如厨房用品，日用小件，清洁工具这三者在这个社区中是关联较为紧密的，说明用户喜欢同时购买这些商品小类中的商品，如果其中一个小类的商品促销，可能会潜在性的带动其他商品小类商品的销售。



图5-1 社区3的商品网络

### 5.3.2基于独立级联模型的购物篮分析的实验结果

算法选出影响力最大的15个商品所属的社区如表5-5所示，每行中商品的顺序是商品加入到该社区的影响力最大化商品集合的顺序。从表5-3和表5-5可知社区6和社区7的规模很小，商品的影响也因此变得比较小，所以没有从社区6和社区7选择促销商品。

表5-5 基于独立级联模型的购物篮分析算法选取的商品组合

|  |  |
| --- | --- |
| **社区编号** | **影响最大商品组合** |
| 1 | 肉松, 进口食醋 |
| 2 | 单反相机, 其它办公用品, 电热杯 |
| 3 | 砍骨刀, 漏勺/滤勺, 其它家居, 其它刀具 |
| 4 | 脱毛工具（美体/脱毛）, 须后水, 干燥剂, 雨披 |
| 5 | 婴儿牙胶, 其它身体护理用品 |
| 6 |  |
| 7 |  |

选取的商品具有的特征如表5-6所示。从表5-6可以知道影响力最大的商品的平均销量排名和平局带权入度是很靠后的，而平均带权出度排名在中上位置的特点。结合表5-5和表5-6可以发现，每个社区最先选择的影响最大商品，均是该社区的影响最大的商品中带权出度最靠前的，但带权入度排名是很靠后的。如肉松带权出度排名是9，在社区1的促销商品中，带权出度排名最靠前，但带权入度排名963，带权入度排名很靠后。

表5-6 基于独立级联模型的购物篮分析算法选取的商品特征

|  |  |  |  |
| --- | --- | --- | --- |
| **商品** | **销量排名** | **带权出度排名** | **带权入度排名** |
| 肉松 | 983 | 9 | 963 |
| 进口食醋 | 954 | 411 | 893 |
| 单反相机 | 1020 | 845 | 784 |
| 其它办公用品 | 1021 | 869 | 892 |
| 电热杯 | 1018 | 843 | 650 |
| 砍骨刀 | 904 | 366 | 748 |
| 漏勺/滤勺 | 883 | 10 | 688 |
| 其它家居 | 684 | 123 | 939 |
| 其它刀具 | 1022 | 523 | 925 |
| 脱毛工具（美体/脱毛） | 1023 | 680 | 938 |
| 须后水 | 969 | 705 | 933 |
| 干燥剂 | 1004 | 2 | 950 |
| 雨披 | 1027 | 817 | 991 |
| 婴儿牙胶 | 922 | 180 | 826 |
| 其它身体护理用品 | 1030 | 847 | 1003 |
| **平均排名** | 964 | 482 | 875 |

该算法寻找出的商品具有平均销量排名和平均带权出度高，但平均带权出度排名低的特点，反映了选择的影响力最大的商品的销量较低和受到邻居的直接影响虽然较低，但是对邻居商品的直接影响强度大。

因为选取的商品受到邻居商品的直接影响强度低，即邻居商品的销售无法推动这些商品的销售，所以这些商品的销量低是合理的。销量低反映了促销的需要，销量低有可能会导致库存的囤积，以致库存成本的提高。如果将这些商品作为促销商品，可能可以帮助减少库存的压力。

选取的商品对邻居商品的直接影响强度大，即能够有效的影响其邻居商品，并通过邻居商品将影响扩散出去。促销这些商品很有可能会让购物者大量购买相关商品，进而带动其他商品的销售。因此本文提出的算法所选择的促销商品具有一定的合理性。

### 5.3.3基于热量传播模型的购物篮分析的实验结果

实验中热量传播模型的参数设置如表5-7所示，其中*n*是图节点个数，*k*是种子数。

表5-7 热量传播模型算法参数表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **参数组** | ***k*** | ***t*** | ***a*** | ***threshold*** | ***H0*** |
| 1 | 15 | 0.6 | 1.0 | 0.35 | n/k |

在表5-7参数设置下，算法选出影响力最大的15个商品所属的社区如表5-5所示，每行中商品的顺序是商品加入到该社区的影响力最大化商品集合的顺序。从表5-3和表5-8可知社区6和社区7的规模很小，商品的影响也因此变得比较小，所以没有从社区6和社区7选择促销商品。

表5-8 基于热量传播模型的购物篮分析算法选取的商品组合

|  |  |
| --- | --- |
| **社区编号** | **影响最大商品组合** |
| 1 | 肉松，料酒，鲜鸡副产品，进口调味粉 |
| 2 | 镜头 |
| 3 | 磨刀石，桌布，其它保鲜盒/饭盒，其它家居 |
| 4 | 干燥剂，脱毛膏/蜡，婴儿护肤用品，童伞 |
| 5 | 烘焙咖啡，五谷/蔬菜饮料 |
| 6 |  |
| 7 |  |

选取的商品具有的特征如表5-9所示。从表5-9可以知道影响力最大的商品具有平均销量排名和平局带权入度是很靠后的，而平均带权出度排名在中上位置的特点。结合表5-8和表5-9可以发现，每个社区选取的影响最大商品的带权出度排名有一定波动性，例如社区3按选取影响最大的商品顺序，商品的带权出度排名分别是4，148，3，123。

表5-9 基于热量传播模型的购物篮分析算法选取的的商品特征

|  |  |  |  |
| --- | --- | --- | --- |
| **商品** | **销量排名** | **带权出度排名** | **带权入度排名** |
| 肉松 | 983 | 9 | 963 |
| 料酒 | 156 | 85 | 103 |
| 鲜鸡副产品 | 984 | 12 | 967 |
| 进口调味粉 | 1015 | 246 | 1015 |
| 镜头 | 894 | 884 | 317 |
| 磨刀石 | 940 | 4 | 779 |
| 桌布 | 494 | 148 | 865 |
| 其它保鲜盒/饭盒 | 976 | 3 | 862 |
| 其它家居 | 684 | 123 | 939 |
| 干燥剂 | 1004 | 2 | 950 |
| 脱毛膏/蜡 | 936 | 615 | 936 |
| 婴儿护肤用品 | 224 | 714 | 196 |
| 童伞 | 979 | 1 | 916 |
| 烘焙咖啡 | 964 | 430 | 841 |
| 五谷/蔬菜饮料 | 388 | 549 | 308 |
| **平均排名** | 775 | 255 | 730 |

该算法寻找的影响最大商品具有平均销量排名和平均带权出度高，但平均带权出度排名低的特点，反映了选择的影响力最大的商品的销量较低和受到邻居的直接影响虽然较低，但是对邻居商品的直接影响强度大。这些商品与基于独立级联模型的影响最大化算法选取的影响最大商品总体特点是一样的，所以把该算法选取的影响最大商品作为促销商品同样具有一定合理性。

### 5.3.4对比简析

独立级联模型和热量传播模型的影响最大化算法的相同点是均是挑选对其他商品的有较强的直接影响强度（带权出度），但销量较低和受到邻居商品直接影响强度（带权入度）低的特点。

独立级联模型直观简单，如果用先验知识得到边的权值，那么所选取的影响最大化商品集合基本是可以确定的，而热量传播模型因为有多个可控参数，如时间，传导率，初始温度等，可以根据网络特征进行相应的进行相应的调整，可以得到多个影响最大化商品组合供选择。热量传播模型比独立级联模型更具有灵活性，但是也带来一定的操作复杂性。

## 5.5本章小结

本文选取了四个特征来衡量影响最大化商品的特点，第一个是反映销售量的特征（销量排名），其他三个是反映商品重要性的特征（pagerank值排名和带权出度排名）。但是发现商品销量排名与pagerank值排名具正相关性，所以只选择了销量排名、带权出度排名和带权入度排名来描述商品的特点。

在实验中分析选取影响最大化商品集合前的社区划分情况，可以初步的直观的了解到商品所属的经营小类间的直接影响关系的强度。在社区划分的基础上，分析这个两个影响最大化算法求取的商品的一些特点，发现基于独立级联模型的影响最大化算法和基于热量传播模型的影响最大化算法选出的商品均具有对邻居商品的直接影响较高，但是商品的销量较低（销量排名靠后）的特点。经过分析验证了把这两个算法选取的影响最大商品作为促销商品的合理性。

# 第六章 总结与展望

本文旨在为零售商选择超市促销品提供商品组合建议，促销商品的选择是一个组合最优化的问题，即所选的商品组合要尽可能的带动其他商品的销售。为解决这样的一个问题，笔者了解了购物篮分析的研究现状并学习了相关的理论知识。 在了解基于商品网络的购物篮分析的研究现状后，撰写了本文的第二章基于商品网络的购物篮分析概述。了解了常用的商品网络构建方法后基于apriori算法的思想求取商品的相关性，然后用节点表示商品，在同一频繁项集中的商品认为存在边的关系，商品关联性作为边的权重。

为了寻找求解选择促销商品的问题的方法，笔者通过查阅文献发现在社交网络影响力传播研究中有一个研究热点是影响最大化算法研究，该研究是寻找一部分人，通过这些人可以尽可能的影响到更多人去做某件事，如购买某件商品，接受某个决定，给某人投票等。影响最大化问题和促销商品组合的选取问题均是组合最优化问题，并且促销商品带动其他商品的销售的能力可理解为促销商品对其他商品的影响强度。了解社交网络影响最大化的相关研究后撰写了第三章。

基于这样的启发，笔者想将社交网络影响最大化算法应用到商品网络中，求取影响最大化的商品（促销商品）。在第四章提出了基于商品影响力的购物篮分析算法，该方法包括三个重要的步骤，分别是构建商品网络的构建，对商品影响传播进行建模，以及设计基于社区划分的影响最大化算法求解促销商品。

在真实的交易数据中对本文设计的算法进行了实验探究，发现基于独立级联模型的和基于热量传播模型的购物篮分析算法寻找的商品同样呈现出对邻居节点的影响较强，但是商品的销量较低的特点。通过分析验证了把这些影响最大的商品作为促销商品的合理性。

因为在不同的时间（如不同的季节）消费者的行为会呈现不同的特点，因此影响力最大的商品组合可能也会有所变化，笔者在未来可能会给购物篮数据加上时间窗口，尝试分析商品间的影响程度与时间的关系。

# 附 录

## 独立级联模型的主要代码

基于独立级联模型的影响最大化算法的主代码：

|  |
| --- |
| def max\_influ\_using\_ica\_diffu(subg, k):  mag = init\_mag(subg)  seed\_set = []  for i in range(len(subg)):  seed\_set.append([[],0])  for i in range(k):  print(i," seed")  idx = arg\_max\_mag(mag, seed\_set, subg)  seed\_set[idx][0].append(mag[idx][0])  seed\_set[idx][1] += mag[idx][1]  print(mag)  subgth = subg[idx]  mag[idx] = get\_node\_with\_max\_gain(subgth,seed\_set[idx][0])  repeat\_time = 1000  influ\_sum = 0  for i in range(len(subg)):  print(i)  t1 = time.time()  seed\_set[i][1] = avgiac(subg[i], seed\_set[i][0], repeat\_time)  influ\_sum += seed\_set[i][1]  print(time.time() - t1)  return seed\_set |

独立级联模型的主要代码

|  |
| --- |
| from \_\_future\_\_ import division  from copy import deepcopy # copy graph object  import random  import networkx as nx  import networkx as nx  import time  def findccs\_in\_digraph(di\_g):  # remove blocked edges from directed graph G  E = deepcopy(di\_g)  edge\_rem = [e for e in E.edges() if random.random() > E[e[0]][e[1]]['weight']]  E.remove\_edges\_from(edge\_rem)  ccs = list(nx.strongly\_connected\_components(E))  return ccs  def runiac (di\_graph, seeds):  ''' Runs independent arbitrary cascade model.  Input: G -- networkx graph object  seed -- initial set of vertices  Output: T -- resulted influenced set of vertices (including S)  '''  actived\_nodes = deepcopy(seeds) # copy already selected nodes  # ugly C++ version  i = 0  while i < len(actived\_nodes):  for v in di\_graph[actived\_nodes[i]]: # for neighbors of a selected node  if v not in actived\_nodes: # if it wasn't selected yet  w = di\_graph[actived\_nodes[i]][v]['weight']  if random.random() <= w:  actived\_nodes.append(v)  i += 1  return actived\_nodes  def avgiac(di\_graph, seeds, repeat\_time):  '''  Input:  G -- undirected graph  S -- seed set  Ep -- propagation probabilities  I -- number of iterations  Output:  avg -- average size of coverage  '''  avg = 0  for i in range(repeat\_time):  avg += float(len(runiac(di\_graph, seeds)))  avg /= repeat\_time  return avg  def findccs\_in\_digraph(di\_g):  # remove blocked edges from directed graph G  E = deepcopy(di\_g)  edge\_rem = [e for e in E.edges() if random.random() > E[e[0]][e[1]]['weight']]  E.remove\_edges\_from(edge\_rem)  ccs = list(nx.strongly\_connected\_components(E))  return E,ccs  def reach\_node\_num\_gain(digraph, seed\_set):  digraph,ccs = findccs\_in\_digraph2(digraph)  # 创建节点与社区的映射  node\_to\_com = dict()  com\_have\_nodes = dict()  for i,cc in enumerate(ccs):  com\_have\_nodes[i]=[]  for node in list(cc):  node\_to\_com[node] = int(i)  com\_have\_nodes[int(i)].append(node)  new\_digraph = nx.DiGraph()  for node in list(range(len(ccs))):  new\_digraph.add\_node(int(node))  for edge in digraph.edges():  s = edge[0]  t = edge[1]  s\_to\_com = node\_to\_com[s]  t\_to\_com = node\_to\_com[t]  if s\_to\_com!=t\_to\_com:  # 这里做了一个边的反向，注意  new\_digraph.add\_edge(t\_to\_com, s\_to\_com)  adj\_array = np.zeros([len(new\_digraph), len(new\_digraph)], dtype=np.bool8)  in\_degree = new\_digraph.in\_degree()  zero\_degree\_nodes = []  for key in in\_degree:  if in\_degree[key] == 0:  zero\_degree\_nodes.append(key)  while len(zero\_degree\_nodes)>0:  zd\_n = zero\_degree\_nodes.pop()  adj\_array[zd\_n][zd\_n] = 1  for node in new\_digraph.neighbors(zd\_n):  in\_degree[node] -= 1  if in\_degree[node] == 0:  zero\_degree\_nodes.append(node)  adj\_array[node] = np.bitwise\_or(adj\_array[node], adj\_array[zd\_n])    scores = dict()  for node in digraph:  scores[node] = 0  seed\_set\_reach\_com = np.zeros(len(new\_digraph), dtype=np.bool8)  for node in seed\_set:  seed\_set\_reach\_com |= adj\_array[node\_to\_com[node]]  # 按位取反后消除可达的的节点  seed\_set\_reach\_com = ~seed\_set\_reach\_com  for com in com\_have\_nodes.keys():  adj\_array[com] &= seed\_set\_reach\_com  for com in com\_have\_nodes.keys():  sum\_node = 0  for com2,value in enumerate(adj\_array[com]):  if value == 1:  sum\_node += len(com\_have\_nodes[com2])  for node in com\_have\_nodes[com]:  scores[node] = sum\_node  return scores |

## 热量传播模型的主要代码

基于热量传播模型的影响最大化算法的主代码：

|  |
| --- |
| def max\_influ\_using\_heat\_diffu(k, heat\_function, threshold = 0.6, t = 0.3, alpha=1.0, num = 1):  """  Input：  k ： 需要选取的种子节点个数  t ： 与邻居节点的交互时间  threshold : 激活阈值  num : 影响力传播的层数，总花费时间是 t \* num  output:  seed\_set : 种子集合  """  mag = init\_mag(subg\_H, k, heat\_function, threshold, t, alpha, num)  seed\_set = []  for i in range(len(subg\_H)):  seed\_set.append([[],0])  for i in range(k):  idx = arg\_max\_mag(mag, seed\_set, subg\_H)  seed\_set[idx][0].append(mag[idx][0])  seed\_set[idx][1] += mag[idx][1]  H = subg\_H[idx]  h0 = heat\_function(H.shape[0])  mag[idx] = get\_new\_one\_of\_mag\_value(H, mag[idx], seed\_set[idx], h0, threshold, t, alpha, num)  return seed\_set |

获取影响增益最大的节点的代码

|  |
| --- |
| def get\_node\_with\_max\_gain(H, seed\_set, h0, threshold = 0.6, t = 0.1, alpha = 1.0, num = 1):  """  挑选一个种子， 该的种子 与 种子集合，一起传播的影响力最大  input:  H : 一个子图的影响力传播矩阵  seed\_set : 当前子图已经选择的种子集合， 一个集合{节点1，节点2}  h0 : 种子的初始温度  threshold ： 激活阈值  t : 与相邻节点的交互时间, 可以理解为单位时间  alpha : 传导系数  num : 影响力传播的层数，总花费时间是 t \* num  return :  (idx, node\_influ[idx]) ： (节点， 节点的影响力)  """  number\_node = H.shape[0]  c = np.matrix( np.zeros((number\_node, num)) )  node\_influ = []  for node in range(number\_node):  new\_seed\_set = seed\_set.copy()  new\_seed\_set.append(node)  # get submatrix  select\_index = new\_seed\_set.copy()  for i in range(num):  select\_index = get\_out\_nei\_index\_also\_own(H, select\_index)  submatrix\_size = len(select\_index)  select\_index = sorted(select\_index)  submatrix\_H = get\_sub\_matrix(H, select\_index)  map\_submatrix\_index = { index:i for i,index in enumerate(select\_index)}  # 初始化温度  f0 = np.zeros(submatrix\_size)  for node in new\_seed\_set:  f0[map\_submatrix\_index[node]] = h0  ft = f0.copy()  ft = ft[:, np.newaxis]  ll = ft  hh = submatrix\_H  for i in range(num):  ft = h\_pass\_t(submatrix\_H, ft, t, alpha)  sum\_influ = 0  for heat in ft:  if heat > threshold:  sum\_influ += 1  node\_influ.append([node, sum\_influ])  node, influ = max(node\_influ, key = lambda x: x[1])  return (node, influ) |

计算H矩阵的代码

|  |
| --- |
| def get\_subg\_sm(subg):  """  Input  subg :子图列表  return :  subg\_sm ： 每个子图的邻接矩阵，有weight则是weight的值矩阵，否则关联则为1  """  par\_num = len(subg)  # 子图的相似度矩阵  subg\_sm = []  for i in range(par\_num):  subg\_sm.append(nx.to\_numpy\_matrix(subg[i]))  return subg\_sm  def get\_subg\_outd(subg):  """  input:  subg : 子图列表  return:  subg\_d : 每个子图节点的出度数，就是有多少条出边  """  par\_num = len(subg)  # 子图的相似度矩阵  subg\_outd = []  for i in range(par\_num):  d\_dict = subg[i].out\_degree()  nodes = subg[i].nodes()  d = []  for i in range( len(nodes) ):  d.append(d\_dict[nodes[i]])  subg\_outd.append(d)  return subg\_outd  def get\_subg\_H(subg\_sm, subg\_outd):  """  Input:  subg\_sm : 每个子图的相似度矩阵  subg\_outd : 每个子图的节点的出度数  return :  subg\_H : 每个子图的影响力传播矩阵    Hij = wj \* pij / dj when (vj, vi) belong to E,  Hij = -(Ti \* wi / di) \* sum(pik) when i == j and (vi, vk) belong to E , and Tii = 1 if degree of vi more than 0, otherwise Ti = 0  Hij = 0 otherwise  """  par\_num = len(subg\_sm)  # 求影响力传播矩阵  subg\_H = []  for i in range(par\_num):  nodes\_num = subg\_sm[i].shape[0]  ci = []  for d in subg\_outd[i]:  if d > 0:  ci.append(1.0/d)  else :  ci.append(0)  H = np.matrix( np.zeros((nodes\_num, nodes\_num)) )  # H = np.zeros((nodes\_num, nodes\_num))  for nodei in range(nodes\_num):  # (n x 1) .x (n x 1) => (n x 1)  H.A[nodei,:][:,np.newaxis] = np.multiply( subg\_sm[i][:,nodei] , np.matrix( np.array(ci)[:,np.newaxis] ) )  # H[nodei,:][:,np.newaxis] = np.multiply( subg\_sm[i][:,nodei] , np.array(ci)[:,np.newaxis] )  for nodei in range(nodes\_num):  H[nodei, nodei] = 0  for nodei in range(nodes\_num):  if subg\_d[i][nodei] != 0:  H[nodei, nodei] = -np.sum(subg\_sm[i][nodei, :]) \* 1.0 \* ci[nodei]  subg\_H.append(H)  return subg\_H |

在mag列表获取取影响增益最大节点在mag列表的索引的代码

|  |
| --- |
| def arg\_max\_mag(mag, seed\_set, subg\_H):  """  Input:  mag : [[子图1中节点k，在该子图的影响力增益], [子图2中节点m，在该子图的影响力增益处]... ]  return :  返回索引  """  influ = [ [i,int(item[1]+0.3), len(subg\_H[i]) / (len(seed\_set[i][0]) + 1)] for i, item in enumerate(mag)]  print(influ)  idx, influ, priori= max(influ, key=lambda x : [x[1],x[2]])  return idx |

初始化mag列表

|  |
| --- |
| def init\_mag(subg\_H, k, heat\_function, threshold = 0.6, t = 0.1, alpha = 1.0, num = 1):  """  Input  subg\_H : 子图的传播矩阵列表  k : 需要选取的种子个数  threshold ： 激活阈值  t : 与相邻节点的交互时间  alpha : 传导系数  num : 影响力传播的层数，总花费时间是 t \* num  return：  mag : [[子图1中节点k，在该子图的影响力增益], [子图2中节点m，在该子图的影响力增益处]... ]  存放的每个子图带来最大影响力的节点及其影响力增益  """  mag = []  for subgth in range(len(subg\_H)):  H = subg\_H[subgth]  number\_node = H.shape[0]  h0 = heat\_function(number\_node)  node\_with\_max\_gain = get\_node\_with\_max\_gain(H, [], h0, threshold, t, alpha, num)  mag.append(node\_with\_max\_gain)  return mag |

## 社区划分的主要代码

快速层次聚类主函数代码代码：

|  |
| --- |
| def generate\_di\_dendrogram(di\_graph, part\_init=None, weight='weight', resolution=1., randomized = False):  """Find communities in the graph and return the associated dendrogram  A dendrogram is a tree and each level is a partition of the graph nodes.  Level 0 is the first partition, which contains the smallest communities,  and the best is len(dendrogram) - 1. The higher the level is, the bigger  are the communities  Parameters  ----------  graph : networkx.DiGraph  the networkx graph which will be decomposed  part\_init : dict, optional  the algorithm will start using this partition of the nodes. It's a  dictionary where keys are their nodes and values the communities  weight : str, optional  the key in graph to use as weight. Default to 'weight'  resolution : double, optional  Will change the size of the communities, default to 1.  represents the time described in  "Laplacian Dynamics and Multiscale Modular Structure in Networks",  R. Lambiotte, J.-C. Delvenne, M. Barahona  Returns  -------  dendrogram : list of dictionaries  a list of partitions, ie dictionnaries where keys of the i+1 are the  values of the i. and where keys of the first are the nodes of graph  Raises  ------  TypeError  If the graph is not a networkx.Graph  See Also  --------  best\_partition  Notes  -----  Uses Louvain algorithm  References  ----------  .. 1. Blondel, V.D. et al. Fast unfolding of communities in large  networks. J. Stat. Mech 10008, 1-12(2008).  .. 2. Community structure in directed networks E. A. Leicht1 and M. E. J. Newman1, 2  Examples  --------  >>> G=nx.erdos\_renyi\_graph(100, 0.01, directed=True)  >>> dendo = generate\_di\_dendrogram(G)  >>> for level in range(len(dendo) - 1) :  >>> print("partition at level", level,  >>> "is", partition\_at\_level(dendo, level))  """  if type(di\_graph) != nx.DiGraph:  raise TypeError("Bad graph type, use only directed graph")  # special case, when there is no link  # the best partition is everyone in its community  if di\_graph.number\_of\_edges() == 0:  part = dict([])  for node in di\_graph.nodes():  part[node] = node  return part  current\_graph = di\_graph.copy()  di\_status = DiStatus()  di\_status.init(current\_graph, weight, part\_init)  di\_status\_list = list()  \_\_one\_level(current\_graph, di\_status, weight, resolution, randomized)  new\_mod = \_\_di\_modularity(di\_status)  partition = \_\_renumber(di\_status.node2com)  di\_status\_list.append(partition)  mod = new\_mod  current\_graph = induced\_di\_graph(partition, current\_graph, weight)  di\_status.init(current\_graph, weight)  while True:  \_\_one\_level(current\_graph, di\_status, weight, resolution, randomized)  new\_mod = \_\_di\_modularity(di\_status)  if new\_mod - mod < \_\_MIN:  break  partition = \_\_renumber(di\_status.node2com)  di\_status\_list.append(partition)  mod = new\_mod  current\_graph = induced\_di\_graph(partition, current\_graph, weight)  di\_status.init(current\_graph, weight)  return di\_status\_list[:] |

计算有向图Q值的python 代码

|  |
| --- |
| def \_\_di\_modularity(di\_status):  """  Fast compute the modularity of the partition of the graph using  status precomputed  """  links = float(di\_status.total\_weight)  result = 0.  for community in set(di\_status.node2com.values()):  internals = di\_status.internals.get(community, 0.)  in\_degree = di\_status.in\_degrees.get(community, 0.)  out\_degree = di\_status.out\_degrees.get(community, 0.)  if links > 0:  # eii - ai\_out \* ai\_in  result += internals / links - in\_degree \* out\_degree / (links \*\* 2)  return result |

将每个节点到调整至一个相邻的社区，加入这个社区可达模块值增益最大。

|  |
| --- |
| def \_\_one\_level(di\_graph, di\_status, weight\_key, resolution, randomized):  """Compute one level of communities  """  modified = True  nb\_pass\_done = 0  # 计算当前的模块值  # https://en.wikipedia.org/wiki/Modularity\_(networks) 公式（4）  cur\_mod = \_\_di\_modularity(di\_status)  new\_mod = cur\_mod  while modified and nb\_pass\_done != \_\_PASS\_MAX:  cur\_mod = new\_mod  modified = False  nb\_pass\_done += 1  nodes = di\_graph.nodes()  if randomized:  random.shuffle(nodes)  for node in nodes:  # 节点所属的社区  com\_node = di\_status.node2com[node]  degc\_totw\_in = di\_status.in\_gdegrees.get(node, 0.) / di\_status.total\_weight degc\_totw\_out = di\_status.out\_gdegrees.get(node, 0.) / di\_status.total\_weight  neigh\_communities = \_\_di\_neighcom(node, di\_graph, di\_status, weight\_key)  \_\_di\_remove(node, com\_node,  neigh\_communities.get(com\_node, 0.), di\_status)  best\_com = com\_node  best\_increase = 0  # key  for com, dnc in neigh\_communities.items():  incr = resolution \* dnc - \  di\_status.in\_degrees.get(com, 0.) \* degc\_totw\_out - \  di\_status.out\_degrees.get(com, 0.) \* degc\_totw\_in  if incr > best\_increase:  best\_increase = incr  best\_com = com  \_\_di\_insert(node, best\_com,  neigh\_communities.get(best\_com, 0.), di\_status)  if best\_com != com\_node:  modified = True  new\_mod = \_\_di\_modularity(di\_status)  if new\_mod - cur\_mod < \_\_MIN:  break |

# 

# 参考文献

# 外文资料

## 外文文献1和译文

**A product network analysis for extending the market basket analysis**

**Abstract**

In this study, we propose a product network analysis, a network-based analysis to analyze a network leveled relation among all products. Compared to market basket analysis, which focuses on the transaction-leveled relation between products, the suggested product network analysis focuses on extended network-leveled point of view of the relation between all products. For such a purpose, we suggest two kinds of product networks, market basket networks and co-purchased product networks. Two networks are comparatively evaluated to analyze the topological characteristics and the structure of those networks. The extended use of market basket analysis, network-leveled analysis are expected to be more effectively and efficiently used in personalized services, such as cross selling, up selling, and personalized product display utilizing the deep relation between products.

Keywords: Market basket analysis; Product network; Social network analysis; Personalized service

1. **Introduction**

Web business has been growing rapidly keeping the pace with the Internet. Thanks to rapid advancement and wide distribution of portable client equipment such as iPhone—Apple’s multimedia and Internet-enable mobile phone—and other smart phones, the mobile Internet technology has becomes more applicable leading to the growth of the Web business market (Agrawal & Srikant, 1994). Its rapid growth has made both companies and customers face a new situation. The companies in the Web business domain, such as web retailers have become to be harder to survive due to more and more competitions.

On the other hand, the customers have confronted with product overload where the consumer is no longer able to effectively choose the products he or she is exposed to, resulting in a poor purchase decision making in a given time interval. As a result, the need for new marketing strategies such as one-to-one marketing, Web personalization, and customer relationship management (CRM) has been stressed both from researches as well as from practical affairs (Sarwar, Karypis, Konstan, & Riedl, 2000a; Sarwar, Karypis, Konstan, & Riedl, 2000b). One-to-one marketing attempts to improve the nature of traditional marketing by using information technology to assist businesses in treating each customer individually. Web personalization is a marketing related action that tailors the Web experience to a particular customer, or set of customers. CRM pursues to acquire new customers, retain existing customers, and grow customer profitability through understanding customers and strengthening relationships with them.

For such a purpose, web retailers have invested heavily in extracting important information from their vast customer databases and product feature databases. Market basket analysis (also known as association rule mining) is one of the data mining methods (Berry & Linoff, 2004; Sarwar et al., 2000a, 2000b) focusing on discovering purchasing patterns by extracting associations or co-occurrences from a store’s transactional data. Each customer purchases a different set of products, in different quantities at different times. Market basket analysis finds product set which is purchased together in a same basket (Agrawal & Srikant, 1994; Chen, Tang,

Shen, & Hu, 2005; Hou & Micheline, 2006; Silverstein, Brin, & Motwani, 1998). Therefore, it is very useful for retailers to use such information, for example, to arrange products, and to make catalogs for up-selling or cross-selling. These purchase patterns are represented in the form of association rules. Market basket analysis has been intensively used in many companies as a means to discover product associations and base a retailer’s promotion strategy on them. Market basket analysis is often used to analyze the transaction-leveled relation among the products purchased together at the same time. However, market basket analysis cannot handle the relation among the products which are purchased together in a period of time, but not necessarily purchased at a same time.

The past few years, many researchers have used network analysis to study complex systems in a wide variety of scientific, social and engineering domains. Examples include World Wide Web (Romualdo, Alexei, & Alessandro, 2001), organization network (Davis, Hossain, & Murshed, 2007), and software development process (Wu, Goh, & Tang, 2007). Watts and Strogatz (1998) represented the collaborative relation between movies and actors by bipartite network containing two types of nodes. Similarly, sales transactions are represented by customer-product bipartite networks, and customer network extracted from customer-product bipartite network have been studied in recommendation system-related researches (Kim, Kim, & Ryu, 2009a, 2009b; Zan, Zeng, & Chen, 2007). Product network is also extracted from customer-product bipartite network, but it has not been much studied yet.

Raeder and Chawla (2009) take a network-analytic approach to the market basket problem, treating it as a search for relationships rather than associations. They build a social network out of the individual products in the store, (hereafter called a product network) and use community detection algorithms to isolate strong relationships. They study structural characteristic of a product network, its degree distribution, to suggest ways in which the composition of a product network differs from that of a traditional social network. Finally, they introduce a novel utility measure for communities of products and show that ranking communities by this measure can quickly isolate important relationships. They evaluate their methods both comparatively and financially, and conclude that community detection provides a valuable supplement to association rules (Raeder & Chawla, 2009). Even though their research is regarded as a first approach of a network-leveled analysis for the MBA, their research focused only on the relationships of the network, and communities of products.

In this study, we extended the MBA into a network level, and propose a product network analysis which can analyze the relationship between products which are purchased together. The suggested network-based analysis focuses on network-leveled global point of view of the relation between all products, which is not limited to the association analysis of market baskets. For such a purpose, we propose two kinds of product networks in this study. One product network named as market basket network (MBN) is based on market basket analysis. Market basket network spatially expands the relationship between some products purchased together into the relationship among all products using network analysis. Another product network named as co-purchased product network (CPN) is extracted from customer-product bipartite network that is extracted from the sales transaction data (Kim et al., 2009a, 2009b; Zan et al., 2007). It analyzes the relation between products purchased by same customers. Therefore, this network expands the relation among products from the products purchased by same customers and at the same time into the products purchased by same customers and not necessarily at the same time.

We compared MBN and CPN by analyzing the characteristics, structure of two networks and some products which plays a central role in two networks. The extended use of market basket analysis and network-leveled analysis are expected to be used in more effective and efficient personalized services, such as one-to-one marketing, Web personalization, or CRM utilizing the deep relationship between products. For example, we can use market basket network to analyze product information to provide appropriate products for customers when there are no customer-related information; and when we want to analyze the customer purchase preference in the long term, we can use co-purchased product network to find what most customers prefer. This study may supply more useful information for retailers, and to customize better products or services for customers. We constructed MBNs and CPNs from customer-product bipartite network, and compared their topological characteristics and performances.

**扩展购物篮分析的一种商品网络分析**

**摘要**

在这项研究中，我们提出了一种商品网络分析方法，一种用网络分析方法分析一个网络所有商品间的关系。与购物篮分析这种关注在交易记录层面上的商品关系的方法相比，建议的商品网络分析方法关注扩展到网络层面视角来看待所有商品之间的关系。为了这样的目的，我们提出了两种商品网络，购物篮网络和同时购买的商品的网络。这两种网络对对拓扑特征和结构进行了对比评价。购物篮分析的扩展使用，网络层面的分析，预计会更有效和准确的被用在个性化服务中，例如交叉销售，向上销售，和利用了商品之间更深层次的关系的个性化产品展示。

关键词：市场购物篮分析；网络产品；社交网络分析；个性化服务

**1．介绍**

随着互联网的发展，网络业务迅速增长。由于快速发展和分布广泛的便携式客户端设备如iPhone——苹果的多媒体和可联网的手机和其他智能手机，移动互联网的技术已经变得更加适用，导致Web业务市场的增长（Agrawal和Srikant，1994）。它的快速增长使企业和客户都面临新的形势。网络商务领域的公司，如网络零售商，由于越来越大的竞争而变得越来越难生存。

另一方面，消费者面临的产品过多，消费者不再能够有效地选择他或她自己想要的商品，导致在给定的时间间隔出现很少的购买决策。因此，从研究以及从实际事务，新的营销策略是一直被强调的需要，如一对一营销，个性化服务，客户关系管理（CRM）（Sarwar, Karypis, Konstan, & Riedl, 2000a; Sarwar, Karypis, Konstan, & Riedl, 2000b）。一对一营销尝试通过使用信息技术来改善传统营销的本质，以帮助企业单独处理每个客户。网络是一个个性化营销的相关行动，为一个特定的客户，或客户集调整Web体验。客户关系管理寻求获得新的客户，保留现有的客户，并通过了解客户和加强与他们的关系发展客户的盈利能力。

为了这样一个目的，网络零售商已经投入巨资从他们庞大的客户数据库和商品特征数据库中提取重要信息。市场篮子分析（也被称为关联规则挖掘）是数据挖掘的方法之一（Berry & Linoff, 2004; Sarwar et al., 2000a, 2000b）主要是从一个商店的交易数据中提取关联性或同时出现发现购买模式。每一个客户购买在不同的时间会购买不同数量的不同的产品。购物篮分析发现在同一个篮子里一起购买的商品集（Agrawal和Srikant，1994；Chen，Tang，Shen，&Hu，2005，Hou&Micheline，2006；Silverstein，Brin，&Motwani，1998）。因此，使用这样的信息对零售商来说是非常有用的，例如，安排商品，并制作向上销售或交叉销售的目录。这些购买模式用关联规则的形式表示。购物篮分析法已被广泛应用于许多公司，作为一种手段，发现商品的关联性和零售商的促销策略以购物篮分析为基础。购物篮分析法常被用来分析在交易层面的，在同一时间购买的商品之间的的关系。然而，购物篮分析法不能处理在一段时间内购买的产品之间的关系，并且这些商品没必要在同一时间购买。

在过去的几年中，许多研究人员已经使用网络分析，在各种各样的科学研究复杂系统，社会和工程领域。例子包括万维网（Romualdo, Alexei, & Alessandro, 2001)），组织网络（Davis, Hossain, & Murshed, 2007），和软件开发过程（Wu, Goh, & Tang, 2007）。Watts和Strogatz（1998）用含两类节点二分网络表示电影和演员之间的合作关系。同样，销售交易用客户产品的二分网络表示，和客户网络是从客户商品的二分网络提取的，在推荐系统的相关研究中被研究（Kim, Kim, & Ryu, 2009a, 2009b; Zan, Zeng, & Chen, 2007）。产品网络也从客户产品的二分网络中提取，但它并没有太多的研究。

Raeder and Chawla (2009)将网络分析法用在市场篮子的问题，把它作为一个关系搜索而不是关联性。他们在商店里建立个人商品的社交网络（以下称为商品网络），并使用社区检测算法来隔离有强关系的组。他们研究了产品的网络结构特点，其度分布，表明其产品网络的组成与传统的社交网络的不同方式。最后，他们介绍了一种用于商品社区的新颖的实用方法，这项措施展示通过排名社区可以快速隔离重要的关系。他们对比地和经济地评估了他们的方法，并得出结论，社区发现是关联规则的一个有价值的补充（Raeder & Chawla, 2009）。尽管他们的研究被认为是MBA的网络层次分析的第一种方法，他们的研究仅集中在网络和社区的商品的关系。

在这项研究中，我们扩展MBA到网络水平，并提出一个商品网络分析用来分析一起购买产品之间的关系。提出的基于网络的分析侧重于在整个网络的视图来看所有商品之间的关系，这不局限于购物篮的关联分析。出于这样的目的，我们这研究中提出了两种产品网络。名字为购物篮网络（MBN）的商品网络是基于购物篮分析的。购物篮网络空间上从一些商品扩展到整个网络，并用网络分析所有产品之间的关系。名字为一起购买的产品网络（CPN）是另一个在从销售交易数据中提取的客户—商品二分网络中提取的商品网络（Kim et al., 2009a, 2009b; Zan et al., 2007）。它分析了同一客户购买的商品之间的关系。因此，网络扩展由同一时间到由同一客户购买的商品关系，到同一客户购买的商品和不必在同一时间购买的关系。

我们比较了MBN和CPN，通过分析特征，网络结构与部分在网络中心的商品。购物篮分析的推广使用和网络层次分析，预计会更有效和准确的被用在个性化服务中，如一对一营销，Web个性化，或利用产品之间更深度的关系的客户关系管理。例如，当没有客户相关的信息时，我们可以用购物篮网络分析为客户提供适当产品；当我们要分析客户长期的购买偏好，我们可以使用共同购买的产品网络找到大多数客户的偏好。这项研究可以提供更多的对零售商有用的信息，并为更好的产品定制或为客户服务。我们从客户—产品二分网络构建了MBNS和CPNS，并比较它们的拓扑结构特性与性能。

## 外文文献2和译文

**Efficient Influence Maximization in Social Networks**

**ABSTRACT**

Influence maximization is the problem of finding a small subset of nodes (seed nodes) in a social network that could maximize the spread of influence. In this paper, we study the influence maximization problem from two angles in order to significantly reduce the running time of existing algorithms. One is to improve the original greedy algorithm of and its improvement, and the second is to propose new degree discount heuristics for the problem. We evaluate our algorithms by experiments on two large academic collaboration graphs obtained from the online archival database arXiv.org. Our experimental results show that (a) our improved greedy algorithm achieves better running time comparing with the improvement of with matching influence spread, (b) our degree discount heuristics achieve much better influence spread than classic degree and centrality-based heuristics, and when tuned for a specific influence cascade model, it achieve almost matching influence thread with the greedy algorithm, and more importantly (c) the degree discount heuristics run only in milliseconds while even the improved greedy algorithms run in hours in our experiment graph with a few tens of thousands of nodes.

**社交网络中高效的影响力传播**

**摘要**

影响最大化问题是在社交网络中寻找一个能够最大化影响力传播的小的节点子集(种子节点)。在这篇论文，我们从两个角度研究影响最大化问题为了显著的减少现存的算法的运行时间。一是改善原有的贪心算法及其改进，二是在这个问题上提出新的度折扣（degree discount）启发式的方法。我们通过从在线数据库存档获得arXiv.org两个大型学术合作图表实验评价我们的算法。我们的实验结果表明，（a）在匹配的影响力扩散，我们的改进贪心算法的运行时间比改进的要好，（b）当调整为具体的影响级联模型，我们的度折扣启发算法式比经典基于度和基于中心性的启发式算法能达到更好的影响传播。（c）在我们有了几万个节点的实验图中，当甚至是改进的贪心算法运行需要几个小时时候，基于度折扣启发只运行几毫秒。

## 外文文献3和译文

**Mining Social Networks Using Heat Diffusion Processes for Marketing Candidates Selection**

**ABSTRACT**

Social Network Marketing techniques employ pre-existing social networks to increase brands or products awareness through word-of-mouth promotion. Full understanding of social network marketing and the potential candidates that can thus be marketed to certainly offer lucrative opportunities for prospective sellers. Due to the complexity of social networks, few models exist to interpret social network marketing realistically. We propose to model social network marketing using Heat Diffusion Processes. This paper presents three diffusion models, along with three algorithms for selecting the best individuals to receive marketing samples. These approaches have the following advantages to best illustrate the properties of real-world social networks:

(1) We can plan a marketing strategy sequentially in time since we include a time factor in the simulation of product adoptions;

(2) The algorithm of selecting marketing candidates best represents and utilizes the clustering property of real-world social networks; and

(3) The model we construct can diffuse both positive and negative comments on products or brands in order to simulate the complicated communications within social networks. Our work represents a novel approach to the analysis of social network marketing, and is the first work to propose how to defend against negative comments within social networks. Complexity analysis shows our model is also scalable to very large social networks.

**在社交网络中使用热量传播挖掘市场的候选人**

**摘要**

社交网络营销技术，采用预先存在的社交网络，通过口碑传播来增加品牌或产品的知名度。充分了解社交网络营销和潜在的候选人，因此可以被市场营销中未来的卖家提供有利可图的机会。由于社交网络的复杂性，很少有模型解释真实的社交网络营销。我们建议使用热扩散过程模型的社会网络营销。本文提出了三种扩散模型，以及三种算法，根据营销样本选择最佳的一个。这些方法有以下优点，最能说明现实世界的社交网络的属性：

（1）我们可以按时间顺序依次计划营销策略，因为我们的方法在模拟商品采用时包含了一个时间因子；

（2）选择最佳营销候选人的算法，并利用现实世界社交网络的聚类特性；

（3）我们所建构的模式，可以传播对产品或品牌的正面和负面的评论，以模拟社会网络内的复杂通讯。我们的工作代表了一种新的方法来分析社交网络营销，并且是第一个提出如何抵御社会网络负面评论的工作。复杂性分析表明，我们的模型也可扩展到非常大的社交网络。

## 外文文献4和译文

**Building Mini-Categories in Product Networks**

Abstract

We constructed a product network based on the sales data collected and provided by a Fortune 500 Specialty Retailer. The structure of the network is dominated by small isolated components, dense clique-based communities, and sparse stars and linear chains and pendants. We used the identified structural elements (tiles) to organize products into mini-categories—compact collections of potentially complementary and substitute items. The mini-categories extend the traditional hierarchy of retail products (group–class–subcategory) and may serve as building blocks towards exploration of consumer projects and long-term customer behavior.

Keywords: retailing, product network, mini-category, category management

**在商品网络中构建微小类别**

摘要

我们根据财富500强的专业零售商提供和收集的销售数据构建产品网络。网络的结构是以孤立的小部件，基于密集派，社区和稀疏的星星，直链和吊坠为主。我们使用确定的结构元素来组织商品到一个微小的类别，一个紧凑的互补和替补商品的集合。微小类别扩展传统的零售商品的层次结构(组—类—子类)和可能作为探索消费项目和客户的长期行为的积木。

关键词：零售，产品网络，迷你品类，品类管理

# 致 谢

论文写至此也意味着即将毕业，回想大学的生活基本是工作室，食堂和寝室3点一线，收获不算颇丰，相对于刚入学的茫然少年已然成长不少。

记得刚上大一的时候，对于计算机这个名词的直觉反应是看电影，网游戏，写写word文档等。初入大学心中不禁有些茫然和强烈的新鲜感，有幸遇到了acm训练招小伙伴就一头扎入编程的世界。在acm训练过程中，我知道了玩电脑除了看电影和玩游戏，通过编程解决实际问题也是一件有趣的事情，并且在这里认识了一路支持我和鼓励我的朋友们，还有一位将我领进科研这个大门的充满热情的杨书新老师。

杨老师让我有机会跟王师姐学习社交网络影响最大化理论，在这个过程中学习了如何去阅读一篇论文和通过编程解决实际的问题，现在依然能清晰的记得与王师姐讨论问题，杨老师指导我书写邮件与论文作者交流等情景。这个过程无疑是我最充实和难忘的经历，这也是我选择这个论文题目的原因。一来是我想对所学知识进行一个总结，二来是解决购物篮分析这个我没接触过的问题，可以让我从查找文献，做实验，到最终的书写论文的每一步骤实践一遍。

我衷心的感谢杨书新书新老师一直鼓励着我前进，培养了我对科研的兴趣，耐心的指导我查阅文献，书写论文，还给予我机会参加学术的活动等等。如果不是杨书新老师及时的指导，恐怕我会在大学的学习和生活中走很多的弯路。在此，向老师表示由衷的敬意和真诚的感谢！

感谢吴雅林老师在大三和大四期间指导我学习图像处理的知识，以及让我了解了正在兴起的虚拟现实研究，并给我们分享了在韩国的科研经历。与吴老师交流的过程中，不管是在生活还是科研都教了我们很多。

感谢陈小军老师给予我去深大做毕设的机会和为我提供购物篮这个应用场景，并及时的指出我的实验存在的问题并给予我建议。在这个过程我培养了主动学习的习惯，参加了学术活动并开阔了科研的视野。

感谢工作室的师兄师姐在我大学成长过程中给予我很多的建议，帮助我解开读研的一些疑惑。感谢一直督促和鼓励我备战考研和推免的游捷同学，陪我度过愉快大学生活的室友，陪我奋战多天完成论文的丁美坤同学等等，大学遇见你们真好。

最后，我想感谢我的家人，特别是我的父母，一直给予我无微不至的关怀和支持。父母是我在成长路上的坚强的后盾,给予了我源源不断的动力，没有你们的陪伴就没有我现在的成就。

感谢一路陪伴我的所有亲人和朋友，在此致以诚挚的谢意！

1. [] Julander C. BASKET ANALYSIS: A NEW WAY OF ANALYSING SCANNER DATA[J]. Inte-rnational Journal of Retail & Distribution Management, 1992, 20(7), 139-153. [↑](#endnote-ref-1)
2. [] Agrawal R. Mining association rules between sets of items in large databases [J]. Acm Sigmod Record, 1993, 22(2):207-216. [↑](#endnote-ref-2)
3. [] Chena, Y. L., Tang, K., Shena, R. J.et al. Market basket analysis in a multiple store environm-ent[J]. Decision Support Systems, 2005, 40(2), 339-354. [↑](#endnote-ref-3)
4. [] Berry, Michael J A, Linoff, et al. Data mining techniques for marketing, sales and customer re-lationship[M], 2004. [↑](#endnote-ref-4)
5. [] Tang K, Chen Y L, Hu H W. Context-based market basket analysis in a multiple-store environ-ment[J]. Decision Support Systems, 2008, 45(1):150-163. [↑](#endnote-ref-5)
6. [] Haughton, D., J. Deichmann, A. Eshghi, et al. A review of software packages for data mining[J]. The American Statistician., 2003, 57 (4): 290-309. [↑](#endnote-ref-6)
7. [] Pingho T, Pan S, Shuoshiung C. Finding ideal menu items assortments: an empirical applicat-ion of market basket analysis.[J]. Cornell Hospitality Quarterly, 2010, 51(4):492-501. [↑](#endnote-ref-7)
8. [] Bell D R, Corsten D, Knox G. Unplanned Category Purchase Incidence: Who Does It, How Often, and Why[J], 2009, 4(1822) : 172-186. [↑](#endnote-ref-8)
9. [] Hoanca, B. & Mock, K. Using Market Basket Analysis to Estimate Potential Revenue Increa-ses for a Small University Bookstore[C]. Conference for Information Systems Applied Research, 2011. [↑](#endnote-ref-9)
10. [] Chandra B, Bhaskar S. A new approach for generating efficient sample from market basket data[J]. Expert Systems with Applications, 2011, 38(3):1321-1325. [↑](#endnote-ref-10)
11. [] Woo J. Market Basket Analysis algorithms with MapReduce[J]. Wiley Interdisciplinary Reviews Data Mining & Knowledge Discovery, 2013, 3(6):445–452. [↑](#endnote-ref-11)
12. [] Kim H K, Kim J K, Chen Q Y. A product network analysis for extending the market basket analysis[J]. Expert Systems with Applications, 2012, 39(8):7403–7410. [↑](#endnote-ref-12)
13. [] 戚威. 基于复杂网络的购物篮商品网络分析研究[D]. 江苏科技大学, 2014. [↑](#endnote-ref-13)
14. [] Zinoviev D, Zhu Z, Li K. Building Mini-Categories in Product Networks[J]. Studies in Computational Intelligence, 2015, 597(2):179-190. [↑](#endnote-ref-14)
15. [] Kempe, David, Kleinberg, Jon, Tardos, et al. Maximizing the spread of influence through a social network[C] ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2003, 88(5):137-146. [↑](#endnote-ref-15)
16. [] Leskovec J, Krause A, Guestrin C, et al. Cost-effective outbreak detection in networks[C] ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2007, 8(2):420-429. [↑](#endnote-ref-16)
17. [] Chen W, Wang Y, Yang S. Efficient influence maximization in social networks[J]. KDD. Proc of Acm Kdd, 2009, 10(2)199-208. [↑](#endnote-ref-17)
18. [] B Borgs C, Brautbar M, Chayes J, et al. Maximizing Social Influence in Nearly Optimal Time[J]. Computer Science, 2012, 77(9):77-95:. [↑](#endnote-ref-18)
19. [] Li H, Bhowmick S S, Sun A. CINEMA: Conformity-Aware Greedy Algorithm for Influence Maximization in Online Social Networks[C] International Conference on Extending Database Technology. 2013, 145(3):323-334. [↑](#endnote-ref-19)
20. [] Ma H, Yang H, Lyu M R, et al. Mining social networks using heat diffusion processes for marketing candidates selection[C]. International Conference on Information and Knowledge Management. 2008, 185(3)::233-242. [↑](#endnote-ref-20)
21. [] Chen Y C, Zhu W Y, Peng W C, et al. CIM: Community-Based Influence Maximization in Social Networks[J]. Acm Transactions on Intelligent Systems & Technology, 2014, 5(2):529-544. [↑](#endnote-ref-21)
22. [] 余颖. 购物篮分析在网络零售业中的应用研究[D]. 天津大学, 2007. [↑](#endnote-ref-22)
23. [] W. Chen, Y. Wang, , and S. Yang. Eﬃcient inﬂuence maximization in social networks[C]. In KDD, 2009, 78(9):420–429. [↑](#endnote-ref-23)
24. [] Ma H, Yang H, Lyu M R, et al. Mining social networks using heat diffusion processes for marketing candidates selection[C] International Conference on Information and Knowledge Management. 2008, 178(2):233-242. [↑](#endnote-ref-24)
25. [] Jiang Q, Song G, Cong G, et al. Simulated Annealing Based Influence Maximization in Social Networks[C]. AAAI Conference on Artificial Intelligence. 2011. [↑](#endnote-ref-25)
26. [] Wang C, Chen W, Wang Y. Scalable influence maximization for independent cascade model in large-scale social networks[J]. Data Mining & Knowledge Discovery, 2012, 25(3):545-576. [↑](#endnote-ref-26)
27. [] Leicht E A, Newman M E J. Community Structure in Directed Networks[J]. Physical Review Letters, 2008, 100(11):2339-2340. [↑](#endnote-ref-27)
28. [] Blondel V D, Guillaume J L, Lambiotte R, et al. Fast unfolding of communities in large networks[J]. Journal of Statistical Mechanics Theory & Experiment, 2008, 30(2):155-168. [↑](#endnote-ref-28)
29. [] Schloegel K, Karypis G, Kumar V. Parallel static and dynamic multi-constraint graph partition-ing[J]. Concurrency & Computation Practice & Experience, 2002, 14(3):219-240. [↑](#endnote-ref-29)
30. [] Schloegel K, Karypis G, Kumar V. Parallel static and dynamic multi-constraint graph partition-ing[J]. Concurrency & Computation Practice & Experience, 2002, 14(3):219-240. [↑](#endnote-ref-30)