|  |
| --- |
| 江 西 理 工 大 学  本 科 毕 业 设 计（论文）  题 目：基于商品影响力分析的购物篮方法研究  学 院：信息工程学院  专 业：计算机科学与技术(计算机软件)  班 级：121班  学 生：袁国文  学 号：20123151  指导教师：杨书新 职称：副教授 |

摘要

随着经济的发展，越来越多的人会去中大型超市购物，超市的数量也在增加，也意味着超市竞争压力的增加。在超市的运营过程，会积累大量的购物篮数据。为了提高顾客的满意度和超市利润，零售商非常关注如何利用购物篮数据来刻画用户的购物行为。目前对购物篮分析的一种常用方法是关联规则分析，然而关联规则得到的是用户购买一些商品再购买另外一些商品的可能性。国外有很少的文献研究利用商品网络来分析购物篮数据，并且这些文献是直接对商品网络进行表面分析，度数和边的分布等。当前仍没找到利用社交网络的影响力分析的方法对商品网络进行分析的应用研究，所以本文会使用经典的影响力分析方法来进行这个探究性实验。

该文将利用两种经典的影响力最大化方法对商品网络进行分析，利用基于热量传播模型和快速层次聚类的影响力最大化算法和基于独立级联模型和层次快速层次聚类的影响力最大化算法对商品网络进行分析，并对找到的影响力最大化的商品的特征进行分析。具体所做的工作如下：

（1）关联规则分析的一种常用方法是apriori算法。本文利用apriori算法，求取商品之间的关联性。以商品作为节点，以关联性作为有向边，那么就可以得到一个有向图。

（2）这一步是为求取影响力最大的商品做铺垫，为了避免所有的影响力最大化商品都集中在一个小社团中，并且在这里我们具有较强关联性的商品会在同一个社团中，商品影响力只会在所属的社团中传播，在不同的社团中商品不会相互影响。本文将无向图的快速层次聚类扩展到有向图中，并设计算法的伪代码，并通过实验分析社团的一些特点。

（3）热量传播和影响力的传播具有一定的相似性，且热量传播模型中时间参数可以在一定程度上模仿促销的临时性。所以本文将设计基于热量传播模型的影响力最大化方法的算法，并从商品成团中选出影响力前K大的商品。

（4）独立级联模型是社交网络中常用的影响力传播模型之一，该模型简单直观，在影响力最大化中表现出良好的特性，如子模特性，方便扩展模型和进一步分析。所以本文将设计基于独立级联模型的影响力最大化方法的算法，并在商品网络的多个社团中寻找影响力前K大的商品。

（5）实验均使用一个大超市在1月份的交易记录作为实验数据，实验的算法实现用的是python语言，结果分析中图表绘制用python，网络绘制使用gephi软件。整个实验包括数据筛选，关联规则计算，社团划分和对上述两种影响力最大化算法的得到的结果进行对比和分析。

**关键字：**购物篮；关联规则；商品网络；社团划分；影响力最大化

ABSTRACT

最后在补充

目录

[基于商品影响力分析的购物篮方法研究 1](#_Toc449712888)

[摘要 1](#_Toc449712889)

[第一章 绪论 1](#_Toc449712890)

[1.1 课题来源及研究背景： 1](#_Toc449712891)

[1.2 国内外研究现状 2](#_Toc449712892)

[1.3 研究内容、意义、思路及架构 3](#_Toc449712893)

[1.4 本文主要创新点 4](#_Toc449712894)

[第二章 本文的相关理论 5](#_Toc449712895)

[2.1网络的定义 5](#_Toc449712896)

[2.2 网络的性质 5](#_Toc449712897)

[2.3 社交网络影响力传播模型 7](#_Toc449712898)

[2.4 影响力最大化问题定义 7](#_Toc449712899)

[2.5 关联规则算法 7](#_Toc449712900)

[第三章 基于购物篮的商品网络分析 8](#_Toc449712901)

[3.1 数据筛选 8](#_Toc449712902)

[3.2 构建商品网络 8](#_Toc449712903)

[3.3 关联规则算法设计： 9](#_Toc449712904)

[3.4 商品网络的特点 10](#_Toc449712905)

[第四章 基于有向图快速层次聚类的商品社团划分 10](#_Toc449712906)

[4.1 社区发现 11](#_Toc449712907)

[4.2 层次聚类算法 11](#_Toc449712908)

[4.3 商品网络社团划分 14](#_Toc449712909)

[第五章 基于社交网络影响力最大化的购物篮方法研究 14](#_Toc449712910)

[5.1 影响力传播模型 14](#_Toc449712911)

[5.2 影响力最大化问题的定义 16](#_Toc449712912)

[5.3 商品网络K个影响力最大化商品的发现 17](#_Toc449712913)

# 基于商品影响力分析的购物篮方法研究

## 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上上的队伍，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所发现，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构建商品网络，并新颖的用社交网络的影响力最大化算法从商品网络寻找影响力延展度最大的前K个商品。

### 1.1 研究背景与意义：

购物篮分析（MBA）是一种广泛使用的技术，可以用来分析顾客同时购买的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等人提出了在购物篮发现关联规则的算法[[2]](#endnote-2)，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（Chen等人 [[3]](#endnote-3)2005；Berry和Linoff等人[[4]](#endnote-4) 2004）。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一起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（tang等人[[5]](#endnote-5)2008，Haughton 等人[[6]](#endnote-6)2003等）。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力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

因基于影响力最大化的购物篮分析，研究较少，目前未能找该方面的研究文献，

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。二是社交网络影响力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问题的研究现状

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反映商品，而边可以用来反映商品之间的关联性。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。除此，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[[13]](#endnote-13)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13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[[14]](#endnote-14)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Zinoviev D等人[[15]](#endnote-15)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13对商品网络的进行简单的度数统计到2015年Zinoviev D等人的抽取商品小类别。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 社交网络影响力最大化的研究现状

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。

影响力最大化（influence maximization）问题一个被广泛研究的影响力传播优化问题。影响力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[[16]](#endnote-16)指出在独立级联模型和线性阈值模型下,影响力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[[17]](#endnote-17)发现在独立级联模型和线性阈值模型下，影响力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力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30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。2013年，Hui Li等人[[18]](#endnote-18)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。2014年Borgs C等人[[19]](#endnote-19)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。

除了基于独立级联模型和线性阈值模型的影响力最大化算法， 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力最大化方法，该方法分为3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。虽然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。使用社交网络影响力最大化方法在商品网络中寻找影响力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力最大化的方法，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。并且可以根据社区特点帮助分析存在较强的相互影响关系的商品类别。
3. 本文选取独立级联模型和热量传播模型作为商品影响力的传播模型，并利用贪心算求解影响力最大化的问题。这两种影响力最大化方法有其各自的特点，本文将探究这两种方法在实际应用的表现。
4. 使用真实的购物篮数据分别进行基于独立级联模型和热量传播模型的影响力最大化实验，对这两种算法实验结果进行评估。

1.3.1 研究内容

（1）了解商品数据，并清洗数据。

（2）在阅读相关文献的基础上，寻找求取商品关联性的方法，并构建商品网络。

（3）学习复杂网络的一些理论知识，并借助复杂网络分析软件分析商品网络。

（4）学习社区分析的一些方法，对商品网络进行社区划分，并分析每个社区的特点。

（5）学习影响力分析的一些方法，探究影响力传播模型在商品网络中的表现。并利用影响力最化方法从中选取影响力最大的K个商品。

1.3.2 研究意义

随着经济的发展，超市的数量也在增加，一方面超市每天的交易量是很大的，尤其是节假日。另一方面超市超市数量的增加，说明超市的竞争也越大，能获得顾客喜爱的超市才能在激烈的竞争中生存下来。

在日常生活中，我们可以看到，超市通过各种促销活动，调整商品摆放等，这些都是精心为顾客带来更多的便利以及引导顾客消费。那么问题来了，促销的时候选择那些商品商品来进行促销呢？顾客的购买习惯是怎样的呢，如何进行商品的摆放才能给顾客带来更多的便利？

本文尝试根据顾客的购买记录构建一个商品网络，每件商品是网络的节点，边表示商品间存在关联，边权表示商品关联性强弱。利用社区划分，分析那些商品的联系更加紧密，可以帮助调整商品的摆放。利用影响力最大化分析，尝试找出影响力最大的商品，可以给商品促销提供更多的信息。

1.3.3 技术路线

本文研究的主要目的是应用复杂网络分析的思想以及社交网络影响力分析的思想对商品网络进行建模分析。本文技术路线如下：

（1）数据清洗

理解原始数据各个字段的含义，抽取所需的字段并进行数据清洗。可视化数据的分布直观理解数据的特征。

（2）构建商品网络，并了解该商品网络的特征。

使用apriori的思想求取两两商品之间的关联性，然后构建起一个商品网络。使用复杂网络的思想，分析该网络的特征。

（3）对超市购物篮网络进行社团划分。

2008年Vincent等人提出了无向图的快速层次聚类。本研究因为使用的是有向图，将结合Leicht等人使用有向图聚类思想，推导出有向图的快速层次聚类。该步骤将研究商品社团的一些特点。

（4）尝试寻找影响力最大的K件商品

探究与传播时间有关的热量传播模型和经典的独立级联模型在商品网络中的表现，并利用贪心算法求取影响力最大的K件商品。

### 1.4 本文组织结构

本文的正文本分全文分为六章，具体组织结构如下：

第一章：绪论。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，接着总结本文所用方法的国内外研究现状，最后阐述本文的主要研究内容和组织结构

第二章，基于商品网络的购物篮分析概述

，如何构建商品网络，商品网络的特点。

第三章， 社交网络影响力最大化概述

本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，相关统计特性，影响力传播模型，影响力最大化问题定义。本章作为知识铺垫。

第四章，基于有向图快速层次聚类的商品社团划分

本章首先介绍社区划分的理论知识，包括传统的层次聚类思想，Vincent等人提出的无向图的快速层次聚类，以及如何应用到有向图中。在理论知识支撑下，对商品网络进行社团划分和分析社团的特点。

第四章，基于独立级联模型的影响力最大化方法的购物篮分析

第五章，基于热量传播模型的影响力最大化方法的购物篮分析

第六章，数据实验与结果分析

本章将会介绍数据的筛选和清洗

购物篮数据特点

本章将会介绍社交网络的两个影响力传播模型，与时间有关的热量传播模型，经典的独立级联模型。还会介绍影响力最大化问题的定义，并且探究影响力最大化方法在求取K个影响力最大化商品的适用性。

第七章：总结与展望

1.4 本文主要创新点

（1）利用商品网络和社团划分从整体的角度研究购物篮特点

传统的购物篮方法大多基于关联规则的，该方法只能得到商品之间的直接关系，不适用于从整体上分析购物篮的特点。国外虽然有使用商品网络来分析购物篮特点的文献，如2012年Hyea Kyeong Kim等人34，但是基于社团划分的购物篮分析方法较少。

本研究将基于关联规则构建商品网络，进而使用快速层次聚类进行划分，从整体上去了解商品的特点。

（2）利用社交网络影响力最大化的思想挑选影响力最大的K件商品

社交网络影响力最大化的研究现在是社交网络分析的一个热点，理论研究也比较成熟，但是在商品网络中仍没找到相应的应用研究文献。在本文中将探究社交网络影响力最大化方法在商品网络中的适用性。

## 第二章 基于商品网络的购物篮分析概述

### 2.1 引言

购物篮分析是零售行业的一种重要的数据挖掘工具。这种方法可以帮助零售商挖掘隐藏在购物篮数据下的有价值信息，辅助商品分类，摆放，促销等经营决策。这一章主要介绍基于购物篮分析的定义和应用，以及商品网络的定义，常见的商品网络的构建方法，以及商品网络目前的研究方法。

### 2.2 购物篮分析定义和应用

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. 促销活动的有效果性分析。从购物篮数据中分析以何种策咯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，尝试去满足顾客对手机屏保的需求。
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。

### 2.3 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。在本文中将会用到apriori 算法来构建商品网络，所以在这里提前介绍基础的知识。

#### 2.3.1 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。

|  |  |
| --- | --- |
| 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（xx）计算商品A的支持度：
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（xx）求商品A后购买商品的B的置信度：

1. 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。

#### 2.3.2 apriori的计算过程

**1、支持度计算**

计算项集的支持度的过程是：

|  |
| --- |
| （1）计算含有一个元素的项集出现的频率，然后筛选出support()>= minSupport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support()，然后筛选support()>= minSupport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。

计算项集合间的关联性的过程如下：

|  |
| --- |
| 对每一个  （1）根据公式（xxx）计算，然后筛选出minconfi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minconfi的项集记为。重复这个步骤直到n = k。 |

#### 2.3.3 apriori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xxx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support(B) < minSupport，由公式（xx）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xxx所示红色的节点会被剪枝去掉。



**2、计算关联规则剪枝思想**

求关联规则的剪枝过程如图（xxx）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，={A,B,C}，当。当求BC->A的置信度的时候，如果confident(|) < minConfi，由公式（xx）可知如果A， 那么confident(|) < minConfi，->的置信度就没必要去算了，即如图xxx所示红色的节点会被剪枝去掉。



### 2.4 常见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xxx，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。

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

网络1。2014年戚戚14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xxx的信息可构建的无向商品网络如下：



（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14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xxx为例可转化成图xxx：



（3）网络3。2012年Hyea Kyeong Kim等人13以商品作为节点，如果两个商品在同一天被购买，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13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Zinoviev D等人15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xxx，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。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |

以及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xxx所示，以表xxx构建的网络如图xxx：

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



### 2.5 商品网络分析常见方法

基于商品网络的购物篮分析，常见的分析有两类，一是社团结构发现，二是重要的商品的发现。这里基于2.4所述的商品网络展开介绍方法，因为不是本文的研究重点，所以只介绍基本的思想。

2.4.1 社团发现

1、层次聚类14。

一种非重叠社区发现算法，层次聚类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14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15。

这是一个重叠社区发现的方法，可以寻找出网络中的所有K个节点的完全图。

Zinoviev D等人15使用该方法对2.4节的网络5进行了K-团发现。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。

2.4.2 重要商品的发现

1、pagerank算法14。

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14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）13。

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13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。

第二章 本文的相关理论

2.1网络的定义

网络（图）的简单的形式可用G = (V, E)表示, 其中是节点集合， E是边的集合，所谓的边就是连接两个节点的线。如果边上有权值的话表示带权图，如果边有方向则成为有向图。

当给节点和边赋予不同的含义的时候，图可以表示不同的网络。如果节点是神经元，边是神经连接，则可以表示神经网络。如果节点是物种，边是捕食关系，则可以表示食物链网络。如果节点是人，边表示人与人之间的联系，则可以表示社交网络。

2.2 网络的性质

2.2.1 邻接矩阵

网络的表示方法有很多种，这里使用邻接矩阵来表示。对于一个图G = (V, E),可用邻接矩阵A来表示，其中表示存在一条从节点i和节点j的关联情况。

对于边上没有权值的无网络，

对于边上没有权值的有向网络

2.2.2 权值矩阵

我们将图的边权矩阵记作W，类似于邻接矩阵表示，不同的是节点i与节点j边上的权值，而不是0和1。

2.2.3 度数和带权度数

对于图G = (V, E)，记节点vV，

无向图节点v的度数

deg(j) =

有向图节点v的出度数，入度数，度数分别为：

deg\_out(i) = ，

deg\_in(i) =

deg(i) = deg\_out(i) + deg\_out\_in(i)

如果是一个有向带权图的话，

deg \_with\_weight\_out (i) = ，

deg \_with\_weight\_in (i) =

deg\_with\_weight (i) = deg \_with\_weight\_out (i) + deg\_out \_with\_weight\_in (i)

2.2.4 网络直径

网络中两个节点的距离是两个节点的最短路径长度，记作。而图的直径是网络中任意两个节点最短路径长度的最大值，

2.2.5 有向图的模块值

模块值是网络结构中的一种度量。可以用来度量社团划分的结果，图的模块值越大说明社团内部的连接越紧密，而社团之间的连接越稀疏。

模块值的定义是，给定一些社区，社区内的边减去边随机分布时候边的期望。有向图的模块值[[23]](#endnote-23)如下：

2.3 社交网络影响力传播模型

为了方便阐述，简单的将社交网络定义为有向图G = (V，E，W)，其中V是节点的集合，E是边的集合，W是表的权值。在社交网络中，一个人可以用一个节点来表示，u节点对v节点的影响关系可以用一条边来表示，影响强度用表示。

通常会给每个节点两个可选的状态，活跃状态和非活跃状态。网络中传播的信息，想法或者产品统称实体。那么一个节点接受了相应的实体，我们称该节点出于活跃状态，否则该节点处于非活跃状态。一个节点由非活跃状态变成活跃状态，我们称该节点被激活，他就成为了实体的新传播源。

影响力模型刻画了影响力在网络中的传播模式，包括一个节点的状态如何影响到其他节点的状态以及影响力在网络中如何扩散的。

影响力的传播模型分为很多种，随机模型，博弈论模型，传热病模型和热量传播模型等。随机模型按传播特点有可分为离散时间/连续时间，递进/非递进。传播模型在商品网络中的研究较少，所以本文以探究实验为主，选择了两种经典的传播模型，2010年wei chen等人的基于随机级联的独立级联模型[[24]](#endnote-24)，热量传播模型5。

2.4 影响力最大化问题定义

对于一个给定网络G = (V，E)，影响力传播模型和所需的参数（如激活阈值，边的权值等），影响力最大化就是寻找由个节点组成的种子集合，使得的延展度最大，可表示为。影响力最大化问题属于组合最优化问题，在不同的影响力传播模型下该问题解决难度不同，如果在独立级联模型下，该问题属于的NP难问题，常见的采用贪心算法来逼近最优解。

2.5 关联规则算法

关联规则算法是购物篮分析的一种常用算法，它可以用来分析商品被一起购买的可能性，也可以用来分析购买一些商品后购买另一些商品的可能性。关联规则的一种经典算法是apriori算法，这个算法的剪枝思想是如果S是频繁项集，那么S的任何一个子集也是频繁项集，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。

支持度：项集在数据库中出现的频率。

置信度：在一个项集出现的情况下，另一个项集出现的概率

Apriori计算关联规则主要是利用了条件概率：

P(AB)是AB项集出现的频率，P(B)是B项集出现的频率，A项集B项集的置信度是。

第三章 基于购物篮的商品网络分析

本章将会介绍数据的筛选和清洗，购物篮数据特点，如何构建商品网络，商品网络的特点。

3.1 数据筛选

原始数据中含有的字段有日期，销售流水号，商品ID，商品名称等字段，在这个实验中，并不需要所有的数据，根据时间字段筛选其中的一个月数据，再从中选择所需要的销售流水号，商品小类。以销售流水号作为连接的键，就可以得到每个交易的商品组合了。

这里可以贴上统计的商品小类在交易中出现的次数

3.2 构建商品网络

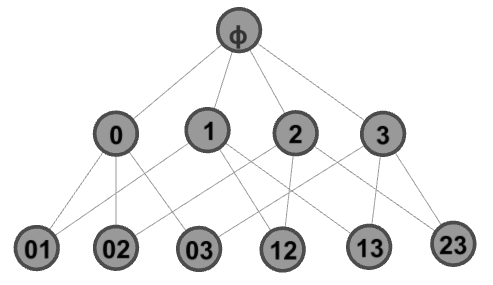
构建商品网络我们将商品小类作为网络的节点，只要商品小类在一次交易中同时出现就认为其存在关系，然后使用apriori算法的条件概率的思想来构建边。

我们可以使用apriori算法来求取项集频率和商品小类的关联规则。使用一个简单的例来理解这个过程，假设有4个商品小类{0, 1, 2, 3}，交易记录如下：

|  |  |
| --- | --- |
| Transaction | Items |
| T1 | 0,1,3 |
| T2 | 1,2 |
| T3 | 1,2,3 |

假设项大小为1和2的项集分别记作L1和L2。

项集出现频率的求取过程是，先求L1出现的频率，然后再将L1组合成L2，求L2的出现频率。



两两的关联规则好求写，知道了L2项集，那么可以将L2项集的每个商品小类集合拆分开，根据条件概率求彼此的关联规则，如{0,1}商品小类集合：

3.3 关联规则算法设计：

求项大小为1的项集伪代码如下：

|  |
| --- |
| Algorithm：createL1(dataSet)  输入：dataSet  输出：项大小为1的项集L1  L1←empty list  for each transaction in dataset  for each item in transaction  if [item] not in C1  insert [item] to C1  输出C1 |

求项大小为K的项集伪代码如下：

|  |
| --- |
| Algorithm：aprioriLn(Lk)  输入: 项大小为K的项集Lk  输出：项大小为K+1的项集Ln  Ln←empty list  lenLk←the length of Lk  n←Lk中项的大小 + 1  for each i in [0,1...lenLk-1]  for each j in [i+1,i+2...lenLk-1]  C1 = list(Lk[i])[:n-2]  C2 = list(Lk[j])[:n-2]  if C1 equal to C2  insert Lk[i]∪Lk[j] to Ln  输出Ln |

计算项集的出现频率伪代码如下：

|  |
| --- |
| Algorithm：scanD(dataset, Lk, minSupport)  输入：交易记录集dataSet，项集Lk，最小支持度minSupport  输出：满足约束条件的项集ret\_Lk，项集的中项出现频率supportData  ItemsetCnt←empty dictionary  for each tid in dataset  for each itemset in Lk  if itemset is not a subset of tid  if itemset is not one of key in ItemsetCnt  ItemsetCnt [ itemset ]←1  else  ItemsetCnt [ itemset ]←ItemsetCnt [ itemset ] + 1  numItems←float(length of dataSet)  ret\_Lk←empty list  supportData←empty dictionary  for each key in ItemsetCnt  support←ItemsetCnt[key] /numItems  if support >= minSupport  insert key to ret\_Lk  supportData[key]←support  输出：ret\_Lk，supportData |

计算任意两个商品小类间的关联规则伪代码如下：

|  |
| --- |
| generateRules(Lk, supportData, minConf)  输入：项集列表L，输入每一项的支持度supportData，最小置信度minConf  输出：关联规则ruleList  ruleList←empty list  len\_Lk←length of Lk  for i in [1...len\_Lk-1]  for each itemset in L[i]  for item in itemset  confidence←supportData[item]/supportData[itemset-item]  if confidence >= minConf  insert (itemset-item, item, confidence) in ruleList  输出：ruleList |

3.4 商品网络的特点

3.4.1 商品网络的直径

3.4.2 商品网络边的特点

3.4.3 商品网络节点度数的分布

3.4.4 商品网络带权度数的分布

## 第三章 社交网络影响力最大化概述

### 3.1 引言

影响力最大化问题是社交网络影响力传播研究的一个热点问题。影响力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、选择影响力最大化求解算法。

因为本文探究社交网络影响力最大化算法在商品网络中的应用，所以在本章以求解影响力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，影响力最大化问题的定义，影响力传播建模，影响力最大化问题研究的关键技术（包括影响力最大化问题求解算法概述和本文求解影响力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E={,|}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G(V,E,W)，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W来表示。与存在影响关系的个体我们称为邻居，用N()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vi节点影响的节点称为节点用，节点节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xxx所示：

表3.2.1 文中通用符号对应含义

|  |  |
| --- | --- |
| 符号 | 含义 |
| G(V,E,W) | 社交网络图 |
|  | G(V,E,W)中的个体集合 |
|  | G(V,E,W)的个体数量 |
|  | G(V,E,W)中第个节点 |
|  | 对出邻居的影响的影响总和（带权出度） |
|  | 入的邻居对的影响总和（带权入度） |
|  | 节点对出邻居的影响 |
|  | G(V,E,W)中的个体间关系集合 |
|  | G(V,E,W)中W中边权的和 |
|  | 编号为的边 |
|  | 以为起点，为终点的边 |
|  | 初始用户集合，或称：种子节点集 |
| ­seed\_num | 中所能包含的最大的节点数量 |
| 或 | 影响传播目标节点 |
|  | 的邻居集合 |
|  | 节点出邻居集合 |
|  | 节点的入邻居集合 |

#### 3.2.2 影响力最大化问题的定义

在给定的网络G=(V,E,W)，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(S)来表示。

影响力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个节点所组合的集合影响力大。我们可用公式xxx来表示：

。

### 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16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

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。让从最初种子集合到第t轮已经被激活的节点，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轮被激活的条件是：

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。

#### 3.3.2 独立级联模型

2003年，Kempe、Kleinberg等人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等人改进了独立级联模型，通过概率1-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图G(V,E,W)，V是节点集，E是边集合，W是边的权值集合，表示从节点到节点的边的权重。让在第t轮被激活的种子集合。 = S表示初始的种子集合。在t>0轮，一个给定的节点V，被它在t-1步骤激活的邻居节点v影响的概率是独立的，其中。在这样的模型下节点u被激活的概率可以用下面的公式来表示：

按照上面的迭代步骤，直到为空。

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。求取强连通组件C,其中。对于一个节点,在的可达节点均在，此时。对于节点V\S，如果节点，那么在选择影响力最大节点时候为了避免选择影响重叠节点，就将的影响力记作0，否则为。独立级联模型求取节点影响力的时候往往需要进行多次模拟取节点的平均影响力。

使用2009年W. Chen等人改进的独立级联模型求节点影响力增益的伪代码如下：

|  |
| --- |
| Algorithm ICM(G, W，S)  输入：图G, 边的权值W，种子节点集合S  输出：影响力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.for r in [1,2,...,N]  4. ←复制G  5. for each 中的边  6. if  7. 从移除  8. C←求的强连通组件  9. For each in V\S  10. if  11.  13. end if  14. end for  15.for in f  16.  17.输出f |

第1~2步骤是初始化，5~7步骤是对独立级联模型的模拟，节点不能通过边成功影响,则移除边。在新的图中，节点可达的节点均可认为是被激活的节点，所以在求影响力前用步骤8求出图的强连通组件。9~14步骤是用于求节点的影响力增益，步骤10中的 就是防止与S节点重叠影响的节点算作节点的影响增益。15~16是将节点平均的影响力增益作为节点的最终影响力。

#### 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（HDM）型进行建模。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，促销多久会带来比较好的结果，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G(V,E,W),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vi传播出去的温度记作DH(i, t, )。

（2）DH(i, t, )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：

其中

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。

当时候有：

求积分后可得：

使用热量传播模型求节点影响力的伪代码如下：

|  |
| --- |
| Algorithm HeatDiffusion(g, S, h0, a, t, ):  input：图g，种子集合S，初始温度h0，传导率a，时间t, 激活阈值  output：影响力  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 v in S  4. ←h0 # 初始化种子节点的温度为h0  5.求解公式xxx得到H  6.计算(t) =  7.for each in g  8. if  9. += 1  10.return |

算法HeatDiffusion中第1~4步骤是初始化，步骤5求解影响力传播矩阵，计算,步骤6求解公式xxx得到每个节点经过时间t后的变化，第7~9步是计算当前被激活的节点数量。

使用热量传播模型求节点影响力增益的伪代码如下：

|  |
| --- |
| Algorithm HeatDiffusionGain(g, S, h0, a, t, ):  input：图g，种子集合S，初始温度h0，传导率a，时间t, 激活阈值  output：影响力增益  1.cur\_incluence←HeatDiffusion(g, S, h0, a, t, )  2.num = g.number\_of\_node  3.f←zeros(num) # 0 数组，元素个数是num个  4.for each node in g  5. if node in S  6. continue  7. new\_seed\_set =  8. influence = HeatDiffusion(g, new\_seed\_set, h0, a, t, )  9. f[node] = cur\_incluence – influence  10.return f |

算法HeatDiffusionGain中第1步种子集合的影响力，步骤2~3初始化每个节点影响力增益为0，步骤4~9是计算每个节点加入种子集会带来的影响力增益

#### 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 影响力最大化关键技术

这一节主要讲述影响力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 影响最大化求解算法概述

影响力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力最大化问题。

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等人指出在独立级联模型和线性阈值模型下,影响力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个节点的影响力最大化种子集。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(1-1/),其中e是常数。

2008年Hao Ma等人[[25]](#endnote-25)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I轮种子集合时候，使用第I-1次求取的种子集S与一起求取影响力，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17发现BasicGreedy贪心在独立级联模型和线性阈值模型求解影响力最大化问题过程中，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[[26]](#endnote-26)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(1-1/)。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。

1. 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[[27]](#endnote-27)使用模拟退火算法来求解影响力最大化问题，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S=，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T=T-，直到T<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[[28]](#endnote-28)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力最大化方法。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3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

本文求影响力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xxx和xxx，以及介绍算法的流程。

给定一个有向图G(V,E,W)，节点的带权出度用表示,带权入度则用来表示，根据2007年E. A. Leicht1等人的文献可知有向图的模块值计算公式为(xxx):

其中

当节点v和节点w在同一个社区的时候，为1否则为0。

引入公式xxx，其中x和y是社区的编号。当x=y的时候。

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

社区x中节点入度和占图总的边权和比例用为表示，则有

社区x中节点的出度和占图总的边权和比例用表示，则有

有向图的模块值可根据公式xxx简化为：

由有向图公式导出一个孤立点加入社区后的模块值的增益：

其中：

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。

有向图的快速层次聚类的伪代码如下：

|  |
| --- |
| Algorithm fhc  input：G,=0.001  output：subg  1.for each len(G.V)  2. cluster[i] #将每个节点看作是一个社区  3.Q2cluster划分状态下，用公式xxx计算模块Q值  4.do  5. for each in G  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xxx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. QQ2  18. Q2  19.while Q2 - 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)  25.return subg |

步骤1~2把每个节点当作一个社区，步骤3计算当前社区划分的模块值。步骤5~16将每个节点加入其中一个邻居社区，加入后带来模块值的增益最大。步骤4~19 不断调整节点到不同社区，直到模块值的增益变化小于值。步骤20~24根据每个社区的节点集合，从原图抽取出相应的子图。步骤25返回子图列表。

第四章 基于有向图快速层次聚类的商品社团划分

4.1 社区发现

社区发现是复杂网络分析的一个经典问题。社区划分是将一个网络划分成多个社区，使社区内的节点联系紧密，而社区间的连接稀疏。将网络的一部分节点划分到一个社区划的质量好坏，可以用模块度量。模块值越大，说明这些节点更可能在一个社区。模块值最大约束下，将社区划分成多个社区的问题是一个NP难问题。社区发现的方式有很多种，比如1970年Kernighan等人[[29]](#endnote-29)基于贪心算法的二分法，1973年Fiedler等人[[30]](#endnote-30)基于谱二分法，2002年Girvan等人[[31]](#endnote-31)通过移除边聚集系数最大的边来划分社区，2004年Newman等人[[32]](#endnote-32)基于模块值的聚类方法。在2008年Blondel等人[[33]](#endnote-33)提出来高效的层次聚类。社区发现在近两年也有很多的研究，2014年，Zhou等人[[34]](#endnote-34)提出了一种结合博弈论和Shapley Value值去评估个体紧密连接的聚类方法。在2016年feiping nie等人[[35]](#endnote-35)提出了一种高效的聚类方式，这种聚类方法能通过不断学习相似度矩阵获得K个簇，而不需用K均值来获得聚类。

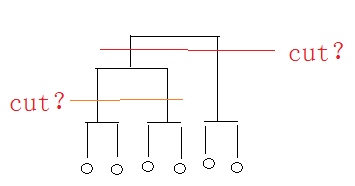
在本文中，依然采用经典的算法，使用2007年Newman等人[[36]](#endnote-36)提出的有向图模块值计算公式，将2008年Vincent D Blondel等人30的无向图快速层次聚类调整成适用于有有向图快速聚类方法。

4.2 层次聚类算法

4.2.1 层次聚类思想

|  |
| --- |
| 1、将每个节点看作是一个社区  2、合并两个距离最近（或相似度最大）的社区为一个新的社区  3、直到最终成为一个社区 |

如下图，层次聚类需要知道在哪里停止聚类，即一个评估聚类好坏的值，有的算法使用模块值来衡量聚类结果，在4.2.3节会讲到该模块值。



4.2.1 无向图快速层次聚类

给定一个无向图，是节点v的带权度，是节点w的带权度，是从v节点和w节点的边的权值。

2004年M. E. J. Newman等人提出的无向图模块值，计算公式如下：

当节点v和节点w在同一个社区的时候为1，否则为0。

根据2004年Aaron Clauset等人的文献对模块值进行简化：

引入：

=，

社区i和社区j连接的边权和与图总的边权和比值为，

社区i中节点的边权和与图总的边权和的比值为，

Q值计算公式可简化为：

一个孤立点i加入社区C的模块值增益：

其中表示当前社区C内的边权和，是节点i连接C内部边权和，是连接到社区C的节点的边权和，m是图的边权值和，无向图 。

在上面的公式基础上，快速层次聚类伪代码如下：

|  |
| --- |
| 1、将每个节点看作是一个社区  2、计算Q值  3、for vertex in Graph:  计算vertex加入与其关联的社区带来的值  选取 最大值，如果则，将节点  加入相应的社区，否则该节点保持原状。  4、计算Q值，比上次状态的Q值变化大于则跳至3  5、否则输出聚类结果 |

4.2.2 有向图快速层次聚类

因为研究的商品网络是有向的，所以需要对算法进行调整，在2007年E. A. Leicht1等人提出了有向图的模块值计算：

给定一个有向图G，表示节点v的带权出度，表示节点w的带权出度，当节点

其中

当节点v和节点w在同一个社区的时候，为1否则为0。

类似于无向图模块值的简化，引入：

社区i和社区j连接的边权和与图总的边权和比值为，

社区i中节点入度和与图总的边权和比值为，

社区i中节点的出度和与图总的边权和比值为，

有向图的模块值可以简化为：

有向图中，一个孤立点加入社区后的模块值的增益：

其中：

表示当前社区C内的边权和，

是节点i连接C内部边权和，

是边的终点是社区C的节点的边权和，

是边的起点是社区C的节点的边权和，

有向图边权和 。

4.3 商品网络社团划分

## 第四章 影响力最大化算法在商品网络中的应用

### 4.1 引言

本章介绍如何将社交网络影响力最大化算法应用到商品网络中，涉及的内容有商品影响力是如何表现的，独立级联模型和热量传播模型如何对商品影响力进行描述的，构建商品网络的伪代码实现，在独立及级联模型和热量传播模型中设计影响力最大化算法求影响力最大的商品集合。

### 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。

根据2.3节apriori算法，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。

伪代码中重要的变量的意义：

|  |  |
| --- | --- |
| dataSet | dataSet是一个二维列表，每一行代表一次交易的商品组合 |
| Lk | Lk是项集的列表，里面的每个项集的元素个数是相同的 |
| L2 | 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项集的伪代码如下：

|  |
| --- |
| Algorithm: createL1  Input: dataSet  output: L1  1.L1←empty list  2.for each transaction in dataset  3. for each item in transaction  4. if [item] not in C1  5. insert [item] to C1  6.return C1 |

dataSet是一个二维列表，每一行代表一次交易的商品组合，步骤2~5是遍历所有商品，并且构建1项集的列表C1，列表中的项集是不重复的。

2、通过k项集构造k+1项集的伪代码如下：

|  |
| --- |
| Algorithm: aprioriLn  input: 项大小为K的项集Lk  output：项大小为K+1的项集Ln  1.Ln←empty list  2.lenLk←len(Lk) # Lk项集的k项集的个数  3.n←len(Lk[0]) + 1 # Lk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. Li ← sorted( list(Lk[i]) ) # 第i个项集的元素升序排列  7. Lj ← sorted( list(Lk[j]) ) # 第j个项集的元素升序排列  8. C1 ← Li[:n-2] # 将Li的前n-2个元素赋值给C1  9. C2 ← Lj[:n-2] # 将Li的前n-2个元素赋值给C1  10. if C1 equal to C2  11. if Lk[i]∪Lk[j] not in Ln  12. add Lk[i]∪Lk[j] to Ln  13.return Ln |

程序输入Lk项集列表，第1~2步骤是初始化标量，Ln是存储k+1项集的列表，lenLk是k项集的个数。第4~11步骤的作用是如果k项集的前k-1个元素相同，那么这两个项集就可以构建出k+1项集。如果构建的k+1项集不在k+1项集列表Ln中，则添加k+1项集到Ln中。

3、计算项集的出现频率伪代码如下：

|  |
| --- |
| Algorithm: scanD(dataset, Lk, minSupport)  input：交易记录集dataSet，项集Lk，最小支持度minSupport  output：满足约束条件的项集ret\_Lk，项集的中项出现频率supportData  1.ItemsetCnt←empty dictionary  2.for each tid in dataset  3. for each itemset in Lk  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\_Lk←empty list  11.supportData←empty dictionary  12.for each key in ItemsetCnt  13. support←ItemsetCnt[key] /numItems  14. if support >= minSupport  15. insert key to ret\_Lk  16.supportData[key]←support  17.return ret\_Lk，supportData |

步骤2~8步是统计所有1项集的出现次数，第11~12步是计算1项集的支持度，并筛选出大于最小支持度的项集。因为是计算两两商品之间的关联性，所以只需要求项集1的支持度就可以了。

4、计算任意两个商品小类间的关联规则伪代码如下：

|  |
| --- |
| Algorithm: generateRules(L2, supportData, minConfident)  input：项集列表L，输入每一项的支持度supportData，最小置信度minConf  output：关联规则ruleList  1.ruleList←empty list  2.len\_Lk←length of L2  3.for each itemset in L2  4. for item in itemset  5. if item not in supportData.keys  or itemset-item not in supportData.keys  continue  6. confidence←supportData[item]/supportData[itemset-item]  7. if confidence >= minConfident  8. insert (itemset-item, item, confidence) in ruleList  9.return ruleList |

步骤8中的itemset-item是集合itemset与集合item的差集。第3~7步骤是求频繁项集的子集与差集的关联性，因为是L2中的每个元素是2项集，所以求得的是两两商品间的支持度。

### 4.2独立级联模型在商品网络的应用

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l-1轮激活的节点尝试一次激活l轮未被激活的节点。每个节点的激活可能性是公式xxxx，其中xxx：

在逛超市的过程中，购物者看到将一件商品加入购物篮子，则认为该商品处于激活状态。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。这样的购买行为与独立级联模型存在一定的相似性，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。而商品的影响力可以定义为，如果消费者在超市购买第一件商品后，在最终购买的商品作为该商品的影响力，商品影响力不代表实际会购买量，只是作为该商品带动销售量的能力。

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力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。

基于3.3.2节所述2009年W. Chen等人的独立级联模型模拟方法，求商品影响力增益的伪代码如下：

|  |
| --- |
| Algorithm ICM(G)  输入：有向图G  输出：影响力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.for r in [1,2,...,N]  4. ←复制G  5. for each 中的边  6. if  7. 从移除  8. C←求的强连通组件  9. For each in V\S  10. if  11.  13. end if  14. end for  15.for in f  16.  17.输出f |

第1~2步骤是初始化，5~7步骤是对独立级联模型的模拟，节点不能通过边成功影响,则移除边。在新的图中，节点可达的节点均可认为是被激活的节点，所以在求影响力前用步骤8求出图的强连通组件。9~14步骤是用于求节点的影响力增益，步骤10中的 就是防止与S节点重叠影响的节点算作节点的影响增益。15~16是将节点平均的影响力增益作为节点的最终影响力。最后返回所有节点的影响增益f。

### 4.3 基于热量传播模型的影响力最大化算法设计

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等人使用公式xxx来计算带权有向网络中种子集合的影响力，

其中公式参数含义

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。

为方便叙述先给出变量的含义表，如表xxx：

|  |  |
| --- | --- |
| g | 有向图 |
| S | 种子集合 |
| h0 | 初始温度 |
| a | 传导率 |
| t | 传播时间 |
|  | 激活阈值 |
|  | 节点影响力列表 |
|  | 节点影响增益列表 |
|  | 节点的影响增益 |

基于3.3.3所述2008年Hao Ma等人的热量传播模型思路，使用热量传播模型求节点影响力的伪代码如下：

|  |
| --- |
| 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.求解公式xxx得到H  6.计算(t) =  7.for each in G.V  8. if  9. += 1  10.return |

算法HeatDiffusion中，第1~4步骤是初始化，步骤5求解影响力传播矩阵，计算,步骤6求解公式xxx得到每个节点经过时间t后的变化，第7~9步是计算当前被激活的节点数量。最后返回种子集合的影响力。

使用热量传播模型求节点影响力增益的伪代码如下：

|  |
| --- |
| Algorithm HeatDiffusionGain(G, S, h0, a, t, ):  input：图G，种子集合S，初始温度h0，传导率a，时间t, 激活阈值  output：影响力增益  1.cur\_incluence←HeatDiffusion(G, S, h0, a, t, )  2.num = g.number\_of\_node  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 |

算法HeatDiffusionGain中第1步种子集合的影响力，步骤2~3初始化每个节点影响力增益为0，步骤4~9是计算每个节点加入种子集会带来的影响力增益。最后返回每个节点的影响增益f。

### 4.4 基于贪心思想的影响力最大化算法

求影响力最大化的商品集合是一个组合最优化问题，本文采用效仿2013年hui li等人的xx算法的贪心思想来进行影响力最大化种子集合的选取。因为本文要做的是应用探究实验，算法在时间复杂度上要求不那么严格，所以使用省略了基于子模性质的延迟更新影响的步骤。

在此基础上，为了使种子节点能更均匀的分布在各个社区，本文会给每个社区进行打分（score），当从两个社区的选取的影响力增益最大的节点，如果影响增益差别不大，则选取分数较高的那个社区的节点。

本文设计的计分函数如公式xxx所示，其中num\_node是社区的节点个数，seed\_num是当前选取的种子数量：

这个贪心算法的框架就能适用于独立级联模型和热量传播模型，所以不按模型分开赘述，算法的流程如图xxx，其中处理1是指将图划分成多个子图，记作subg。处理2是指从子图中寻找影响增益最大的节点更新mag列表，当mag相应位置的节点被选为种子节点，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。



为方便理解伪代码，先给出重要变量及其含义，如表xxx：

|  |  |
| --- | --- |
| 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 |

根据图xxx设计选择k节点影响力最大化种子集合的算法（get\_seed\_set）：

|  |
| --- |
| algorithm: get\_seed\_set (G, k):  input: G, k  output: seed\_set  1. subgfhc(g) # 使用3.4.2的有向图社区划分  2. maginit\_mag(subg) # 初始化mag列表  3. seed\_setempty list # 空种子列表  4. for each i in [0,1,2…,len(subg)]  5. seed\_set.append([]) # 初始化每个社区种子集合为空  6. for i in [0,1,…,k-1]  7. idxarg\_max\_mag(mag, seed\_set, subg)  8. seed\_set[idx][0].append(mag[idx][0]) # mag[idx][0]是影响增益最大种子  9. seed\_set[idx][1]seed\_set[idx][1] +mag[idx][1] # mag[idx][1] # 影响增益  10. subgthsubg[idx]  11. mag[idx]get\_node\_with\_max\_gain (subgth, seed\_set[idx][0])  12. return seed\_set |

get\_seed\_set算法步骤1是使用3.4.3节fnc算法进行社区划分，第2~5步是构建mag列表，初始化各个社区种子集合为空。步骤6的循环指的是每次循环获得一个种子节点。步骤7~9是从所有社区中选择影响增益最大的种子。步骤10~11是从获取了最大影响增益的种子节点的社区重新选取一个影响增益最大的种子来更新mag列表。最后返回k节点的影响力最大种子集合。

从一个社区中选择一个影响力最大的节点的伪代码的伪代码：

|  |
| --- |
| algorithm: get\_node\_with\_max\_gain  input: G, seed\_set, dfm\_gain  output: 节点node,收益gain  1. f = dfm\_gain (g, seed\_set)  2. node,gainthe node with max gain from f  3. return node,gain |

步骤1是计算传入的图中每个节点的影响增益，步骤2~3是获得和返回影响增益最大的节点和该节点的增益。

初始化mag列表的伪代码：

|  |
| --- |
| 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 |

初始化mag列表，mag[i]是一个元组，元组内容是[社区i的影响增益最大的节点，该节点的增益]。步骤1构建mag列表，步骤2~4构建mag列表，步骤5返回mag列表。

从mag获取一个影响力增益最大的节点所在mag列表的索引：

|  |
| --- |
| def arg\_max\_mag  input: mag, seed\_set, subg  output: idx  1.new\_magempty list  2.for idx in [0,1,2,…, len(mag)  3. nodemag[idx] # node[0]是节点，node[1]是节点的影响力  4. numsubg[idx].number\_of\_node # 社区节点个数  6. score = num / (len(seed\_set[idx]) + 1) # 使用公式xxx计算score值  7. influ = int(node [0]+0.3) # 取整后的种子节点影响增益  8. new\_mag.append([idx, influ, score])  9.idx, influ, scoremax(new\_mag, key=lambda x : [x[1],x[2]])  10.return idx |

步骤1是构造新的列表记为new\_mag，new\_mag存放的每个元素会是一个元祖，元组内容是(节点在mag的位置， 取整后的种子节点影响增益，社区的score值）。

步骤3~8是新建new\_mag。步骤9~10是获取并返回影响增益最大的节点在mag 的位置索引。

## 第五章 实验

### 5.1 引言

该章主要探究上一章的设计的基于独立模型和基于热量传播模的影响力最大化算法在实际数据集的变现。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

#### 5.3.1 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。

#### 5.3.2 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xxx所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 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xxx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：

|  |
| --- |
| 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 |

构建商品网络，使用4.2节的generateRules方法。实验构建网络选择的最小置信度1/tn，支持度均是0.001,其中tn是商品组合数。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，构建的网络节点数是1039，边的数量是282192。

#### 5.3.2 实验评估方式

选出的影响力最大化商品组合，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重要性度量的特征来分析实验结果。在本文选择三个特征来衡量，商品购买量排名，带权出度排名，pagerank值排名。

1、商品购买量排名。用来衡量影响力最大化商品的销量与畅销商品的关系。

2、带权出度排名。用来衡量影响力最大化商品与邻居节点直接关联的程度

3、pagerank值排名。在2014年戚戚14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力最大化商品的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。

### 5.3 实验结果

#### 5.3.1 社区划分结果

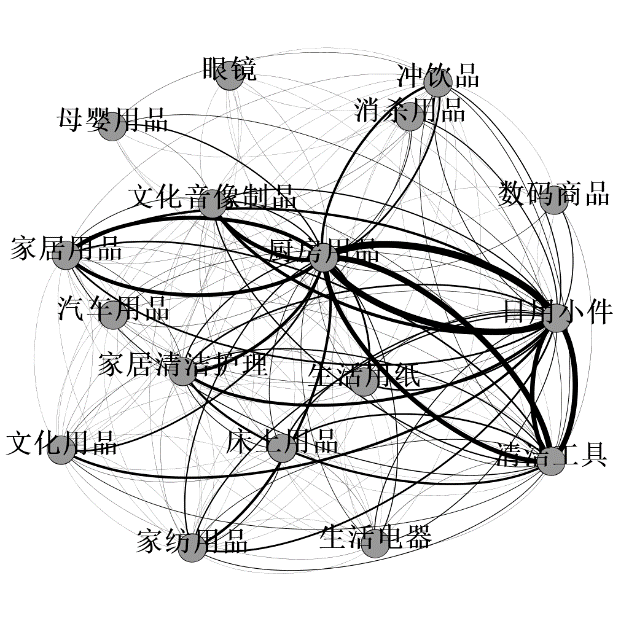
网络被划分成7个社区，特征如表xxx所示：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

每个社区的经营小类数量：

|  |  |  |
| --- | --- | --- |
| 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xxx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。



#### 5.3.2 基于独立级联模型的影响力最大化算法实验结果

实验设置独立级联模型的k为15，选出影响力最大化的商品组合中的15商品所属的社区如表xxx所示：

|  |  |
| --- | --- |
| 社区编号 | 影响最大商品组合 |
| 1 | 冰鲜鱼类 |
| 2 | 其它照相器材，玉兰油 OLAY I |
| 3 | 筷子(竹/木)，汤勺(不锈钢) ，洁厕剂，牙签/牙签盒，陶瓷面碗（5-6寸），其它餐具 |
| 4 | 成人牙刷，杀虫喷雾，有芯卷纸，男士洗面奶，香皂 |
| 5 | 夹心饼干 |
| 6 |  |
| 7 |  |

在整个网络中的商品特征降序得出的排名，筛选出影响力最大组合中的15个商品，如表xxx所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 商品 | 销量排名 | 带权出度排名 | pagerank排名 |
| 冰鲜鱼类 | 65 | 280 | 54 |
| 其它照相器材 | 968 | 866 | 218 |
| 玉兰油 OLAY I | 484 | 858 | 394 |
| 筷子(竹/木) | 310 | 187 | 284 |
| 汤勺(不锈钢) | 593 | 58 | 579 |
| 洁厕剂 | 191 | 139 | 149 |
| 牙签/牙签盒 | 311 | 284 | 311 |
| 陶瓷面碗（5-6寸） | 491 | 142 | 495 |
| 其它餐具 | 260 | 240 | 229 |
| 成人牙刷 | 38 | 480 | 31 |
| 杀虫喷雾 | 675 | 462 | 715 |
| 有芯卷纸 | 24 | 485 | 20 |
| 男士洗面奶 | 328 | 490 | 249 |
| 香皂 | 197 | 451 | 195 |
| 夹心饼干 | 84 | 388 | 70 |

根据销量商品网络的特，绘制商品销量排名与带权值出度排名如图xxx所示，绘制商品销量排名与pagerank排名的散点图如图xxx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，在后续的分析中只用销量排名和带权出度排名进行实验分析。

对独立级联模型的影响力最大化商品的特征可以得到如下的结论

1. 由表xxx和表xxx可知，在6社区和7社区过小，在这两个社区并未选取出种子节点，因为在小社区影响力不能像大社区扩展的那么大。
2. 从表xxx知道影响力最大的商品的销量排名范围是65~968，带权出度排名范围是58~866，pagerank值排名范围是54~715，因为组合最优化的限制，影响力最大的商品并未集中在某一排名段，而是尝试在整个网络中寻找影响力最大节点。从表xxx的销量前5商品特征可以看出带权出度与销量并没有直接的关系
3. 从商品在各个社区的分布情况来看，实验中设置的社区的打分没有起到预期的作用，原因是选取影响力最大化种子过程中，很少出现商品影响相等的情况。

#### 5.3.3 基于热量传播模型的影响力最大化算法实验结果

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 参数组 | k | t | a | threshold | H0 |
| 1 | 15 | 0.6 | 1.0 | 0.35 | 500 |
| 2 | 15 | 0.6 | 1.0 | 0.35 | 200 |
| 3 | 15 | 0.6 | 1.0 | 0.35 | n/k |

参数组1设置下，选出影响力最大化的商品组合中的15商品所属的社区如表xxx所示：

|  |  |
| --- | --- |
| 社区编号 | 影响最大商品组合 |
| 1 | 粽子，腌渍蔬菜，火腿类 |
| 2 | 其它电子瓦煲 |
| 3 | 筷笼，米桶，隔水篮 |
| 4 | 补水喷雾，婴儿爽身粉，其它袜 |
| 5 | 其它罐头，卤蛋，果菜肉泥，儿童水壶/水杯，袋泡黑茶 |
| 6 |  |
| 7 |  |

在整个网络中的商品特征降序得出的排名，筛选出影响力最大组合中的15个商品，如表xxx所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 商品 | 销量排名 | 带权出度排名 | pagerank排名 |
| 粽子 | 836 | 13 | 876 |
| 腌渍蔬菜 | 698 | 53 | 731 |
| 火腿类 | 916 | 7 | 935 |
| 其它电子瓦煲 | 786 | 908 | 207 |
| 筷笼 | 868 | 16 | 875 |
| 米桶 | 903 | 20 | 890 |
| 隔水篮 | 810 | 562 | 851 |
| 补水喷雾 | 839 | 35 | 813 |
| 婴儿爽身粉 | 790 | 789 | 878 |
| 其它袜 | 832 | 84 | 877 |
| 其它罐头 | 876 | 207 | 930 |
| 卤蛋 | 454 | 336 | 512 |
| 果菜肉泥 | 819 | 479 | 864 |
| 儿童水壶/水杯 | 849 | 481 | 918 |
| 袋泡黑茶 | 869 | 756 | 922 |

参数组2设置下，选出影响力最大化的商品组合中的15商品所属的社区如表xxx所示：

|  |  |
| --- | --- |
| 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 |  |

在整个网络中的商品特征降序得出的排名，筛选出影响力最大组合中的15个商品，如表xxx所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 商品 | 销量排名 | 带权出度排名 | pagerank排名 |
| 肉松 | 983 | 9 | 995 |
| 料酒 | 156 | 85 | 119 |
| 鲜鸡副产品 | 984 | 12 | 1021 |
| 进口调味粉 | 1015 | 246 | 1014 |
| 镜头 | 894 | 884 | 169 |
| 磨刀石 | 940 | 4 | 926 |
| 桌布 | 494 | 148 | 503 |
| 其它保鲜盒/饭盒 | 976 | 3 | 963 |
| 其它家居 | 684 | 123 | 667 |
| 干燥剂 | 1004 | 2 | 991 |
| 脱毛膏/蜡 | 936 | 615 | 985 |
| 婴儿护肤用品 | 224 | 714 | 261 |
| 童伞 | 979 | 1 | 957 |
| 烘焙咖啡 | 964 | 430 | 945 |
| 五谷/蔬菜饮料 | 388 | 549 | 421 |

参数组3设置下，选出影响力最大化的商品组合中的15商品所属的社区如表xxx所示：

### 5.3.4 对比简析

### 5.5 本章小结

## 第六章 总结与展望

## 附录

### 独立级联模型的主要代码

基于独立级联模型的影响力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\_mag2(mag, seed\_set, subg)  # idx = arg\_max\_mag(mag)  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  # from priorityQueue import PriorityQueue as PQ  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 greedy\_ic\_digraph(digraph, k, r=20):  seed = []  for i in range(k):  print(i)  time2k = time.time()  scores = {v: 0 for v in digraph}  for j in range(r):  print (j,)  ccs = findccs\_in\_digraph(digraph)  for cc in ccs:  for v in seed:  if v in cc:  break  else: # in case CC doesn't have node from S  for u in cc:  scores[u] += float(len(cc))/r  max\_v, max\_score = max(scores.items(), key=lambda key\_value: key\_value[1])  seed.append(max\_v)  print()  print(time.time() - time2k)  return seed |

### 热量传播模型的主要代码

基于热量传播模型的影响力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]  print(mag)  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)    # print(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 |

### 复杂网络分析工具gephi的使用说明

第五章 基于热量传播模型的影响力最大化方法

第五章，基于社交网络影响力最大化的购物篮方法研究

本章将会介绍社交网络的两个影响力传播模型，与时间有关的热量传播模型，经典的独立级联模型。还会介绍影响力最大化问题的定义，并且探究影响力最大化方法在求取K个影响力最大化商品的适用性。

5.1 影响力传播模型

5.1.1 热量传播模型

为了捕捉市场暂时的信息传播，2008年Hao Ma等人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（HDM）型进行建模。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，促销多久会带来比较好的结果，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G = {V,E,P},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vi传播出去的温度记作DH(i, t, )。

（2）DH(i, t, )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：

其中

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。

当时候有：

求积分后可得：

使用热量传播模型求节点影响力的伪代码如下：

|  |
| --- |
| Algorithm HeatDiffusion(g, S, h0, a, t, ):  输入：图g，种子集合S，初始温度h0，传导率a，时间t, 激活阈值  输出：影响力num  num←0  f←zeros array  for each v in S  ←h0  计算f(t) =  for each v in g  if  num += 1  输出：num |

5.1.2独立级联模型

2003年，Kempe、Kleinberg等人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等人改进了独立级联模型，通过概率1-P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图G = (V,E,P)，V是节点集，E是边集合，P是边的权值集合，表示从节点u到节点v的边的权重。让在第t轮被激活的种子集合。 = S表示初始的种子集合。在t>0轮，一个给定的节点uV，被它在t-1步骤激活的邻居节点v影响的概率是独立的，其中。在这样的模型下节点u被激活的概率可以用下面的公式来表示：

按照上面的迭代步骤，直到为空。

u节点只会尝试一次通过激活v节点，因此我们可以换个角度，可以预先通过一个随机函数，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G’。在图G’中，种子集合S可达的节点集合记作。这样子做的好处是，我们可以通过求取强联通组件，得到相互可达的多个节点集合记作RS,表示第i个强联通组件。对于节点v，有。对于节点V\S，如果节点，那么在选择影响力最大节点时候为了避免选择影响重叠节点，就将v的影响力记作0，否则为。

使用2009年W. Chen等人改进的独立级联模型求影响力的伪代码如下：

|  |
| --- |
| Algorithm ICM(G, W)  从图G中移除边当  计算  计算  For each v in V\S  If v    end if  end for |

5.2 影响力最大化问题的定义

影响力最大化问题的定义是，在给定的网络G=(V, E)，影响力传播模型和所需的参数的情况下，从网络中选择K个种子节点作为种子集合，种子集合的影响扩散后使被激活节点数最大，即。

在热量传播模型下，使用贪心算法求影响力最大种子集合伪代码如下：

|  |
| --- |
| Algorithm Enhanced k-Step Greedy Algorithm  输入：图g，初始温度h0，传导率a，时间t, 激活阈值  输出：影响力最大的K节点种子集和影响力influence  for each l in [1…k]  max\_influ -1  vertex = 0  for each v in V\S  influ HeatDiffusion(g, S{v}, h0, a, t, )  if influ max\_influ  max\_influinflu  vertexv  SS{vertex}  Influencemax\_influ  输出：S，influence |

独立级联模型下，使用贪心算法求影响力最大种子集合伪代码如下：

|  |
| --- |
| Algorithm NewGreedIC(G, k)  输入：有向图G，种子数k  输出：种子集合S  S，N  for each I in [1,2…k]  for each v in V\S  for i in [1,2..N]  从图G中移除边当  计算  计算  For each v in V\S  If v    end if  end for  for each v in V\S  S = S{}  end for  end for |

5.3 商品网络K个影响力最大化商品的发现

## 参考文献

1. Julander, C.-R. (1992). Basket analysis: A new way of analysing scanner data. *International Journal of Retail & Distribution Management , 20* (7), 10-18. [↑](#endnote-ref-1)
2. Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining Association Rules between Sets of Items in Large Databases. *ACM SIGMOD International Conference on Management of Data (SIGNOD ’93)* (pp. 207-216). Washington DC: ACM. [↑](#endnote-ref-2)
3. Chen, Y. L., K. Tang, R. J. Shen, and Y. H. Hu. 2005. Market basket analysis in a multiple store environment. *Decision Support* *Systems* 40 (2): 339-54. [↑](#endnote-ref-3)
4. Berry, M. J. A., and G. S. Linoff. 2004. *Data mining techniques—For marketing, sales, and customer support*. 2nd ed. New York:Wiley. [↑](#endnote-ref-4)
5. Tang, K., Y. L. Chen, and H. W. Hu. 2008. Context-based market basket analysis in a multiple-store environment. *Decision Support* *System* 45:150-63. [↑](#endnote-ref-5)
6. Haughton, D., J. Deichmann, A. Eshghi, and S. Sayek. 2003. A review of software packages for data mining. *The American* *Statistician* 57 (4): 290-309. [↑](#endnote-ref-6)
7. Ting, P.-H., Pan, S., & Chou, S.-S. (2010). Finding Ideal Menu Items Assortments: An Empirical Application of Market Basket Analysis. *Cornell Hospitality Quarterly , 51* (4), 492-501. [↑](#endnote-ref-7)
8. Bell, D. R., Corsten, D., & Knox, G. (2009, January). Unplanned Category Purchase Incidence: Who Does It, How Often, and Why. *Knowledge@Wharton* . [↑](#endnote-ref-8)
9. Hoanca, B. & Mock, K. “Using Market Basket Analysis to Estimate Potential Revenue Increases for a Small University Bookstore.” Conference for Information Systems Applied Research, Vol.4, No.1822, 2011. [↑](#endnote-ref-9)
10. Chandra, B., & Bhaskar, S. (2011). A new approach for generating efficient sample from market basket data. *Expert Systems with Applications , 38*, 1321–1325. [↑](#endnote-ref-10)
11. Market Basket Analysis algorithms with MapReduce. Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery 3(6): 445-452 (2013) [↑](#endnote-ref-11)
12. Kim, H.K., Kim, J.K., Chen, Q.Y.: A Product Network Analysis for Extending

    the Market Basket Analysis. Expert Systems with Applications, vol. 39, 7403–7410

    (2012) [↑](#endnote-ref-12)
13. Kim, H.K., Kim, J.K., Chen, Q.Y.: A Product Network Analysis for Extending

    the Market Basket Analysis. Expert Systems with Applications, vol. 39, 7403–7410

    (2012) [↑](#endnote-ref-13)
14. 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-14)
15. Zhu Z, Li K. Building Mini-Categories in Product Networks [J]. Studies in Computational Intelligence, 2015, 597:179-190. [↑](#endnote-ref-15)
16. Kempe D, Kleinberg J M, Tardos É. Maximizing the spread of influence through a social network. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ( K DD ), Washington D C, USA, 2003: 137~146 [↑](#endnote-ref-16)
17. Leskovec J, Krause A, Guestin C, et al. Cost- effective outbreak detection in networks. Proceedings of the 13rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), San Jose, USA, 2007: 420 ~429 [↑](#endnote-ref-17)
18. Li H, Bhowmick S S, Sun A. CINEMA: Conformity-Aware Greedy Algorithm for Influence Maximization in Online Social Networks[C] Proceedings of the 16th International Conference on Extending Database Technology. 2013:323-334. [↑](#endnote-ref-18)
19. Borgs C, Brautbar M, Chayes J, et al. Maximizing social influence in nearly

    Optimal time. Proceedings of ACMSIAM Symposium on Discrete Algorithms

    (SODA), Portland, USA, 2014: 946~957 [↑](#endnote-ref-19)
20. H. Ma, H. Yang, M. Lyu, and I. King. 2008. Mining social networks using heat diffusion processes for marketing candidates selection. In *Proceedings of the 17th ACM Conference on Information and Knowledge* *Management (CIKM’08)*. 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, 2006. [↑](#endnote-ref-22)
23. Community structure in directed networks E. A. Leicht1 and M. E. J. Newman1, 2 （得改） [↑](#endnote-ref-23)
24. **Efficient Influence Maximization in Social Networks** [↑](#endnote-ref-24)
25. H. Ma, H. Yang, M. Lyu, and I. King. 2008. Mining social networks using heat diffusion processes for marketing candidates selection. In *Proceedings of the 17th ACM Conference on Information and Knowledge* *Management (CIKM’08)*. 233–242. [↑](#endnote-ref-25)
26. Li H, Bhowmick S S, Sun A. CINEMA: Conformity-Aware Greedy Algorithm for Influence Maximization in Online Social Networks[C] Proceedings of the 16th International Conference on Extending Database Technology. 2013:323-334. [↑](#endnote-ref-26)
27. Jiang Q, Song G, Cong G, et al. Simulated Annealing Based Influence Maximization in Social Networks.[C] AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, Usa, August. 2011. [↑](#endnote-ref-27)
28. 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-28)
29. KERNIGHAN, B.W. and S. LIN (1970) An efficient heuristic procedure for structure graphs, Bell System Technical Journal, 49, 291–307. [↑](#endnote-ref-29)
30. FIEDLER, M. (1973) Algebraic connectivity of graphs, Czechoslovak Mathematical Journal, 23, 298–305. [↑](#endnote-ref-30)
31. GIRVAN, M. and M.E.J. NEWMAN (2002) Community structure in social and biological networks, Proceedings of the National Academy of Science, 99, 7821–7826. [↑](#endnote-ref-31)
32. NEWMAN, M.E.J. and M. GIRVAN (2004) Finding and evaluating community structure in networks, Physical Review E, 69, 026113, 1–16. [↑](#endnote-ref-32)
33. Blondel, V.D.; Guillaume, J.-L.; Lambiotte, R.; Lefebvre, E. Fast Unfolding of Communities in Large Networks. J. Stat. Mech. Theory Exp. 2008, P10008, 1–11. [↑](#endnote-ref-33)
34. ZHOU, L. and K. LÜ (2014) Detecting communities with different sizes for social network analysis, The Computer Journal. DOI:10.1093/comjnl/bxu087. [↑](#endnote-ref-34)
35. Feiping Nie, Xiaoqian Wang, Michael I. Jordan, Heng Huang. The Constrained Laplacian Rank Algorithm for Graph-Based Clustering. The 30th AAAI Conference on Artificial Intelligence (AAAI), Phoenix, USA, 2016. [↑](#endnote-ref-35)
36. Community structure in directed networks

    ## 致谢

    [↑](#endnote-ref-36)