# 基于商品影响力分析的购物篮方法研究

计算机科学与技术（软件方向）2012级 袁国文

指导老师 杨书新（副教授）

**摘要**： 促销商品的选择问题是购物篮分析中的一个重要的问题。通过分析购物篮研究的国内外研究现状，发现购物篮常用方法是关联规则分析，并且这个方法并不适合求解促销商品选择这一组合最优化问题。商品网络分析是近几年出现的购物篮分析方法，通过了解sigkdd中的文献发现，利用商品网络求解促销商品的选择的研究仍较少。本文提出在商品网络使用基于独立级联模型影响力最大化算法和基于热量传播模型的影响力最大化算法。独立级联模型不考虑影响的积累，且认为刚加入购物篮的商品只会尝试一次激活相关的商品。而热量传播模型用热量在介质的传播来模拟商品影响力的传播，该模型认为商品影响存在积累效应。基于这两种模型的影响力最大化算法均是基于社区划分的贪心算法。最后，本文在真实数据集探究了这两种基于不同影响传播模型的影响力最大化算法选择的促销商品的特点。

**关键词：**购物篮分析；关联规则；商品网络；社团划分；影响力最大化率。

购物篮分析（Market Basket Analysis，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的可能性。1993年Agrawal等人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发现商品间潜在的关系。后来关联规则分析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2008，Haughton 等人2003等）。在2010年，Ting和Steve等人将购物篮的关联分析应用到了餐馆的菜单选择上。

近几年，陆续有研究者，探究基于商品网络的购物篮分析，因为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。

在DBLP数据库中查询sigkdd收录的基于商品网络的购物篮分析文章可知，基于商品网络的购物篮分析的研究仍比较少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。

与此同时，通过了解社交网络影响力传播的研究现状，发现影响力最大问题也是一个求解组合最优化的问题。在本文中将使用其中的两种算法，基于独立级联模型的影响力最大化算法和基于热量模型的影响力最大化算法

1 相关工作

* 1. 商品网络的现状

基于商品网络的研究主要有商品社团发现和商品重要性的评估等。如2012年Hyea Kyeong Kim等人[12]为了能从整体的视角分析商品间的关系，构建了有向带权商品网络，并分析了边权分布，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年戚戚使用层次聚类进行商品社团的挖掘和使用pagerank算法来寻找重要的商品。2015年Zhu Z等人为了补充商品的分类，提出了在商品网络中挖掘微小商品社团的方法。

构建商品网络的方法有很多种，本文采用apriori算法计算商品两两之间的关联性，然后用节点表示商品，关联性强度不小于最小支持度的商品之间用一条有向边表示，边的权值表示关联性的强度。

* 1. 影响力最大化的现状

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(3.1)来表示：

(3.1)

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[[[1]](#endnote-1)]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。2010年wei chen等人[[[2]](#endnote-2)]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[[[3]](#endnote-3)]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力最大化算法， 2008年Hao Ma等人[[[4]](#endnote-4)]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。

影响力最大化算法有很多种，本文的研究针对购物的行为，采用独立级联模型和热量传播模型对商品影响力传播进行建模，最后使基于社区划分的贪心算法求解影响力最大的商品组合(促销商品的组合)。

1. 独立级联模型

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。

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2009年W. Chen等人[23]的独立级联模型模拟方法，求商品影响力增益的伪代码如表4-6。

1. 表4-6 ICM算法伪代码

|  |
| --- |
| Algorithm ICM(G)  Input: *G*，S |

1. 续表4-6 ICM算法伪代码

|  |
| --- |
| 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.*for r in [1,2,...,N]*  4. ←the copy of *G*  5. for each in  6. if  7. remove from  8. ←Remove node in S from  9. node\_gain = zeros array  10. for each in  11．  12. for each in *G.V*  13.  14.for in *f*  15.  16.return *f* |

*ICM*算法输入有向图*G*和种子节点集合*S*，最后输出节点的影响力增益。第一行是初始化影响力传播模拟次数N为2000。第二行新建一个所有元素值为0的数组*f*。第3行至地16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。第4行是复制一份图*G*为。第5行至地7行是随机的从图中移除边。第8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9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10行至第11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12行至第13行统计每次模拟的影响力增益到*f*中。第14行至第15行求每个节点的影响力增益的平均值。第16行返回每个节点的平均影响增益*f*。

3 热量传播模型

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。

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。

基于2008年Hao Ma等人[20]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表4-7所示。

表4-7 *HeatDiffusion*算法为伪代码

|  |
| --- |
| 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.求解公式3.9得到*H*  6.计算  7.for each in *G.V*  8. if  9. += 1  10.return |

算法*HeatDiffusion*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行是初始化影响力为0。*f*初始化为0数组，记录节点被影响的程度。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。

2、使用热量传播模型求节点影响力增益的伪代码如表4-8所示。

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* |

算法*HeatDiffusion*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行计算当前的种子集合的影响力存储到*cur\_incluence*中。第二行是获得图G的节点数。第3行新建一个元素值为0的数组f。第4行至第9行尝试计算的影响力*influence*，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*。

4 基于社区划分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[19]的*CINEMA*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（4.1）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：

(4.1)

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3-9，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。

图4-9 贪心算法流程



1、设计选择k节点影响力最大化种子集合的算法（*get\_seed\_set*）如表4-10所示。

表4-10 *get\_seed\_set*算法伪代码

|  |
| --- |
| algorithm: *get\_seed\_set*  input: *G*, *k* |

续表4-10 *get\_seed\_set*算法伪代码

|  |
| --- |
| output: *seed\_set*  1. *subgFHC(G)*  2. *maginit\_mag(subg)*  3. *seed\_setempty list*  4. for each *i* in *[0,1,2…,len(subg)]*  5. *seed\_set.append([])*  6. for *i* in *[0,1,…,k-1]*  7. *idxarg\_max\_mag(mag, seed\_set, subg)*  8. s*eed\_set[idx][0].append(*  *mag[idx][0])*  9. *seed\_set[idx][1]seed\_set[idx][1] +mag[idx][1]*  10. *subgthsubg[idx]*  11. *mag[idx]get\_node\_with\_max\_gain (subgth, seed\_set[idx][0])*  12.return *seed\_set* |

*get\_seed\_set*算法输入有向图*G,*可种子个数 *k*，最后输出影响力最大的*k*节点种子集合。第1行是对图*G*进行社区划分得到社区列表*subg*。第2行是初始化*mag*列表，该列表存储每个社区当前会带来影响力增益最大的节点，及其增益。第3行是创建一个空的列表*seed\_set*，用于存储每个社区的种子节点。第4行至第5行是对于每个子图，给*seed\_set*添加一个空列表，表示该社区仍未选出种子节点。第6行至第11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*。第12行是返回影响力最大的种子集合。

2、从一个社区中选择一个影响力最大的节点的伪代码如表4-11。

表4-11 *get\_node\_with\_max\_gain*算法伪代码

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

*get\_node\_with\_max\_gain*算法输入有向图*G,* 种子集合*seed\_set,* 计算影响力增益的函数*dfm\_gain，*该函数是*ICM*或者*HeatDiffusionGain*函数，这两个函数除了前面的两个参数，其余参数有默认初始化值。第一行是计算所有节点的影响力增益。第2行是获得影响力增益最大的节点及其影响增益。第3行是返回影响力增益最大的节点及其影响增益。

3、初始化*mag*列表的伪代码如表4-12。

表4-12 init\_mag算法伪代码

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

续表4-12 init\_mag算法伪代码

|  |
| --- |
| 5.return *mag* |

*init\_mag*算法输入子图列表*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*中添加一个元组，*mag[i]*元组用于存储[社区i的影响增益最大的节点，该节点的增益]。第5行返回mag列表。

4、从*mag*获取一个影响力增益最大的节点所在*mag*列表的索引的伪代码如表4-13。

表4-13 arg\_max\_mag算法

|  |
| --- |
| Algorithm: *arg\_max\_mag*  input: *mag, seed\_set, subg*  output: *idx*  1.*new\_magempty list*  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.*idx, influ, scoremax(new\_mag, key=lambda x : [x[1],x[2]])*  10.return *idx* |

*arg\_max\_mag*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行是新建空列表*new\_mag*，该列表将会存放多个3元组，每个3元组的内容是[节点在*mag*的位置， 取整后的种子节点影响增益，社区的*score*值]。第2行至第8行对于*mag*的第*idx*个元素，根据公式(4.1)计算对应社区的得分，然后将[*idx*，*mag*[*idx*]存放的影响力增益，社区的*score*值]存入到*new\_mag*中。第9行获取new\_mag中影响一增益最大的索引，如果有多个，则选择*score*最大对应的那个索引。第10行返回索引。

5 实验分析

1. [] Kempe, David, Kleinberg, Jon, Tardos, &#. Maximizing the spread of influence through a social network[C]// ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2003:137--146. [↑](#endnote-ref-1)
2. [] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In Proc. of the 15th ACM

   Int. Conf. on Knowledge Discovery and Data Mining

   (KDD’09). [↑](#endnote-ref-2)
3. [] 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. 2013:323-334. [↑](#endnote-ref-3)
4. [] 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. 2008:233-242. [↑](#endnote-ref-4)