# 基于共享最近邻的客户交易数据聚类算法

李 遥,荀亚玲
(太原科技大学 计算机科学与技术学院,山西 太原 030024)

**摘 要**:利用客户交易数据聚类分析,可得到更优异的客户细分效果,有助于企业更详实地了解消费者,制定精准的营销策略。PurTreeClust 是一种新型的客户交易数据聚类算法,定义了一种新型的度量方式 PurTree 距离,可以很好地分析处理具有层次树结构的交易数据,但未考虑近邻点的影响,仅将交易树分配到距离最近的聚类中心所属类簇,容易出现错误的交易树分配。该文利用交易树之间的共享最近邻信息,提出一种客户交易数据聚类算法。该算法在聚类分配时,充分利用共享最近邻,首先分配类簇的从属交易树,然后分配类簇的可能从属交易树,实现聚类分配,可发现更加紧凑清晰的类簇,并避免了交易树错误分配,改善了客户细分效果。最后采用 6 个真实客户交易数据集进行实验,验证了该算法的有效性。

**关键词**:聚类;交易数据;客户细分;交易树;共享最近邻
**中图分类号**:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2022)01-0073-06
doi:10.3969/j.issn.1673-629X.2022.01.013

# A Customer Transaction Data Clustering Algorithm Based on Shared Nearest Neighbors

LI Yao,XUN Ya-ling
(School of Computer Science and Technology,Taiyuan University of Science and Technology,Taiyuan 030024,China)

**Abstract**:By clustering analysis of customer transaction data,better customer segmentation effect can be obtained,which is helpful for enterprises to have a more detailed understanding of consumers and develop accurate marketing strategies. As a new clustering algorithm for customer transaction data,PurTreeClust defines a new measurement method,PurTree distance,which can analyze and process transaction data with hierarchical tree structure. However,without considering the influence of neighboring points,only the purchase tree is allocated to the class cluster belonging to the nearest cluster center,so the wrong purchase tree allocation is prone to occur. We propose a clustering algorithm for customer transaction data using the shared nearest neighbors information among purchase trees. The algorithm makes full use of the shared nearest neighbors to achieve cluster allocation. Firstly,the subordinate purchase tree of the cluster is allocated,and then the possible subordinate purchase tree of the cluster is allocated to realize cluster allocation. It can find more compact and clear clusters,avoid the wrong allocation of the purchase tree,and improve the effect of customer segmentation. Finally,experiments on six real customer transaction datasets verify that the proposed algorithm is more effective.

**Key words**:clustering;transaction data;customer segmentation;purchase tree;shared nearest neighbor

## 0 引 言

客户细分是电子商务和零售领域关注的重要内容之一,利用企业积累的海量客户交易数据,分析客户行为,进行合理的客户细分,有助于企业详尽地了解消费者,在激烈的市场竞争中脱颖而出[1-2]。客户细分传统方式是利用客户年龄、性别等一般属性进行客户细分[3-4],但数据收集较难,细分效果并不理想。聚类分析是数据挖掘中的重要方法之一,使同一个簇中的对象尽可能相似,不同簇之间的对象尽可能相异[5-6]。利用客户交易数据聚类分析,得到同一个簇中的客户拥有更相似的消费习惯,获得了更优异的客户细分效果[7]。但客户之间的相似性度量和客户聚类分配等,是客户交易数据聚类分析面临的主要问题。

针对客户细分聚类分析,Kuo 等人[8]提出一种客户细分聚类算法,利用历史交易数据进行客户细分,但细分效果并不理想;Tsai 等人[9]提出一种基于遗传算法的客户细分方法,依据交易行为划分客户簇并给出合适的营销建议;Lu 等人[10]提出一种基于神经网络

**收稿日期**:2021-02-08 **修回日期**:2021-06-09
**基金项目**:国家青年科学基金项目(61602335);山西省自然科学基金(201901D211302);太原科技大学博士科研启动基金项目(20172017)
**作者简介**:李 遥(1993-),男(回),硕士研究生,研究方向为数据挖掘与并行计算;荀亚玲,博士,副教授,研究方向为数据挖掘与并行计算。

的客户细分算法,利用迭代计算减少簇间相关系数,实现客户细分;Hsu 等人[11]提出一种客户交易数据聚类算法,将客户交易数据组织成树形结构,并利用层次聚类进行客户细分;Yu 等人[12]提出一种基于随机子空间技术的客户交易数据聚类算法,获得了比较准确的结果;Holy 等人[13]分析药店交易数据并提出一种基于遗传算法的商品聚类算法;Chen 等人[14]提出一种从客户交易数据中细分客户的 PurTreeClust 聚类算法,将每个客户的交易记录组织成一棵交易树,并定义一种新型的度量方式 PurTree 距离,更好地反映了两棵交易树之间的距离,但在聚类分配过程中,仅将交易树分配到最近的聚类中心点所属类簇,并未考虑近邻点的影响,容易出现错误的聚类结果。

利用客户交易数据聚类分析,正确地进行聚类分配,可获得更加准确的聚类分簇,得到同一个簇中的客户拥有更相似的消费习惯,有利于企业制定更加精准的营销策略[15]。但 PurTreeClust 在聚类分配过程中,仅将交易树分配到最近的聚类中心点所属类簇,容易出现错误的聚类结果。对此,该文提出一种基于共享最近邻的客户交易数据聚类算法。该算法在聚类分配时,考虑到了交易树之间的共享最近邻信息,不会将交易树直接分配给最近聚类中心所属类簇,有效地解决交易树错误分配问题,并改善了客户细分效果。最后采用六个真实的客户交易数据集进行实验,验证了该算法的有效性,并可以发现更加清晰紧凑的客户细分类簇。

## 1 客户细分和聚类分析

客户细分是指企业根据客户之间的相似性程度,将客户划分成不同的群体,同群体内的客户消费需求相近,不同群体内的客户消费需求差异较大。与客户细分传统方式相比,利用客户交易数据聚类分析,能够更客观地反映不同客户群体的消费需求,有利于营销人员制定更精准的营销策略,提升企业效益。参照文献[14,16]的相关概念定义如下:

定义 1(PurTree 距离):设一个大小为 $n$ 的交易树集合,$H(\Phi)$ 表示交易树高度,$C_v(\varphi_i)$ 表示交易树中节点 $v$ 的全部孩子节点,则交易树与交易树之间的 PurTree 距离定义为:

$$d(\varphi_i,\varphi_j)=\sum_{l=1}^{H(\Phi)}\omega_l\sum_{v\in N^l(\varphi_i)\cup N^l(\varphi_j)}\beta_v\left(1-\frac{|C_v(\varphi_i)\cap C_v(\varphi_i)|}{|C_v(\varphi_i)\cup C_v(\varphi_i)|}\right) \tag{1}$$

其中,$\beta_v$ 为节点 $v$ 的权重,公式为:

$$\beta_v=\begin{cases}1 & \text{if} \quad v=\text{root}(\Phi)\\ \dfrac{\beta_\omega}{|C_\omega(\varphi_i)\cup C_\omega(\varphi_j)|} & \text{where} \quad v\in C_\omega(\varphi_i)\cup C_\omega(\varphi_j)\end{cases} \tag{2}$$

其中,$\omega_l$ 是 $l$ 层的权重,公式为:

$$\omega_l=\begin{cases}\dfrac{1-\gamma}{1-\gamma^{H(\Phi)}}\gamma^{l-1} & for \quad \gamma>0 \quad \text{and} \quad \gamma\neq 1\\ \dfrac{1}{H(\Phi)} & \text{for} \quad \gamma=1\\ 1 & \text{for} \quad \gamma=0 \quad \text{and} \quad l=1\\ 0 & \text{for} \quad \gamma=0 \quad \text{and} \quad 1<l\leqslant H(\Phi)\end{cases} \tag{3}$$

定义 2(共享最近邻):对于数据集 $D$ 中的任意点 $i$ 和点 $j$,设点 $i$ 的 $k$ 近邻是 $\Gamma(i)$,点 $j$ 的 $k$ 近邻是 $\Gamma(j)$,则点 $i$ 和点 $j$ 的共享最近邻是它们的公共部分,定义为:

$$\text{SNN}(i,j)=\Gamma(i)\cap\Gamma(j) \tag{4}$$

定义 3(共享近邻相似度):假设点 $i$ 和点 $j$ 是数据集 $D$ 中的任意不同点,它们的共享近邻相似度定义为:

$$\text{Sim}(i,j)=\begin{cases}\dfrac{|\text{SNN}(i,j)|^2}{\sum_{p\in\text{SNN}(i,j)}(d_{ip}+d_{jp})}, & \text{if } i,j\in\text{SNN}(i,j)\\ 0, & \text{otherwise}\end{cases} \tag{5}$$

其中,$d$ 是点 $i$ 和点 $j$ 的距离。

定义 4(局部密度):假设数据集 $D$ 中的任意点 $i$,$L(i)=\{x_1,x_2,\cdots,x_k\}$ 是与点 $i$ 相似度最高的 $k$ 个交易树集合。那么,点 $i$ 的局部密度定义为:

$$\text{den}(i)=\sum_{j\in L(i)}\text{Sim}(i,j) \tag{6}$$

定义 5(分离距离):假设点 $i$ 和点 $j$ 是数据集 D 中的任意不同点,点 $j$ 的局部密度大于点 $i$ 的局部密度,点 $i$ 的分离距离定义如下:

$$\text{sdis}(i)=\min_{j:\rho_j>\rho_i}\left[d_{ij}\left(\sum_{p\in\Gamma(i)}d_{ip}+\sum_{q\in\Gamma(j)}d_{jq}\right)\right] \tag{7}$$

局部密度最高的点的分离距离,是其他所有点中最高的分离距离。

定义 6(从属点):假设点 $i$ 已被分配到簇 A,而点 $j$ 还未被分配,如果点 $i$ 和点 $j$ 满足公式 8,则交易树 $j$ 是簇 A 的从属点。

$$|\text{SNN}(i,j)|\geqslant k/2 \tag{8}$$

定义 7(可能从属点):假设点 $i$ 已被分配到簇 A,而交易树 $j$ 还未被分配,如果点 $i$ 和点 $j$ 满足公式 9,则点 $j$ 是簇 A 的可能从属点。

$$0<|\text{SNN}(i,j)|<k/2 \tag{9}$$

PurTreeClust 算法根据定义(1)计算交易树之间的 PurTree 距离后,先利用 CoverTree 寻找聚类中心所

在层的所有节点集合 $Q$;然后计算集合 $Q$ 中节点的局部密度、分离距离和分离密度;其次筛选集合 $Q$ 中分离密度最大的前 $K$ 个交易树,作为聚类中心;最后将剩余节点分配到距离最近的聚类中心所在的聚类簇,完成聚类。

## 2　共享最近邻与聚类分配策略

尽管 PurTreeClust 可以比较高效地完成客户交易数据的聚类,但 PurTreeClust 在聚类分配时,只是将客户交易树分配给距离最近的聚类中心所属类簇,容易出现错误分配的情况。如图 1 所示,点 1 和点 3 分别是不同类簇的聚类中心,按照 PurTreeClust 的分配思想,因为点 2 距离点 3 更近,因此会将点 2 分配给点 3 所属类簇,但很明显,点 2 应该分配给点 1 所属类簇,出现了错误分配的现象。

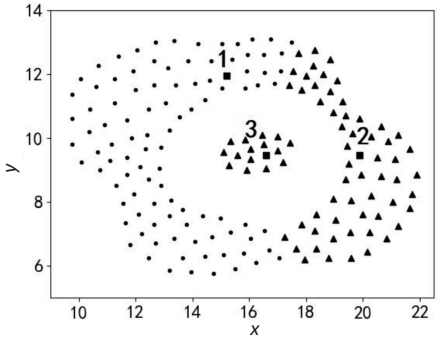



图 1　PurTreeClust 错误分配的情况

出现 PurTreeClust 错误分配的主要原因是在聚类分配时,将客户交易树直接分配给距离最近的聚类中心所属类簇,没有考虑到交易树的近邻影响。在聚类分配时,可考虑到客户交易树之间的共享最近邻信息,不会将客户交易树直接分配给最近聚类中心所属类簇。

由定义 4 可知,局部密度定义为 $\mathrm{den}(i)=\sum_{j\in L(i)}\mathrm{Sim}(i,j)$,其中参照文献[13],共享近邻相似度可定义如下:

$$\mathrm{Sim}(i,j)=\begin{cases}|\mathrm{SNN}(i,j)|\dfrac{1}{\dfrac{1}{|\mathrm{SNN}(i,j)|}\displaystyle\sum_{p\in\mathrm{SNN}(i,j)}(d_{ip}+d_{jp})} \\ \qquad \text{if} \quad i,j\in\mathrm{SNN}(i,j) \\ 0,\text{otherwise}\end{cases} \tag{10}$$

其中,$|SNN(i,j)|$ 表示点 $i$ 与点 $j$ 的共享最近邻个数,$\frac{1}{|\mathrm{SNN}(i,j)|}\sum_{p\in\mathrm{SNN}(i,j)}(d_{ip}+d_{jp})$ 表示点 $i$ 与点 $j$ 与其共享最近邻的平均距离,即当 $|\mathrm{SNN}(i,j)|$ 越大时,点 $i$ 与点 $j$ 之间的共享近邻越多,点 $i$ 周围越密集,点 $i$ 的密度应该越高;当 $\frac{1}{|\mathrm{SNN}(i,j)|}\sum_{p\in\mathrm{SNN}(i,j)}(d_{ip}+d_{jp})$ 越小时,点 $i$ 和点 $j$ 与共享最近邻的平均距离越小,点 $i$ 周围越密集,点 $i$ 的密度也应该越高。由定义 5 可知,分离距离定义为 $\mathrm{sdis}(i)=\min_{j:\rho_j>\rho_i}[d_{ij}(\sum_{p\in\Gamma(i)}d_{ip}+\sum_{q\in\Gamma(j)}d_{jq})]$,与传统分离距离定义相比,定义 5 不仅考虑到距离 $d_{ij}$ 对分离距离的影响,也考虑到了 $k$ 近邻之间的距离影响,即当 $(\sum_{p\in\Gamma(i)}d_{ip}+\sum_{q\in\Gamma(j)}d_{jq})$ 越大时,点 $i$ 和点 $j$ 距离它们的 $k$ 近邻越远,则点 $i$ 和点 $j$ 为低密度区域,低密度区域的点可获得更高的补偿,更加公平地提高了低密度区域中分离距离的准确性。

聚类分配时,首先分配类簇的从属点。由定义 6 可知,类簇的从属点公式为 $|\mathrm{SNN}(i,j)|\geqslant k/2$,即点 $i$ 和点 $j$ 各自的 $k$ 近邻中,有一半以上为两者的共享最近邻,则认为点 $i$ 和点 $j$ 属于同一个类簇,点 $j$ 一定属于点 $i$ 所属类簇。由定义 7 可知,类簇的可能从属点公式为 $0<|\mathrm{SNN}(i,j)|<k/2$,即某未分配点 $j$ 与任意类簇中已分配点 $i$ 的共享最近邻个数满足公式(9)时,则认为点 $i$ 和点 $j$ 有可能属于同一个类簇,即未分配点 $j$ 是已分配点 $i$ 所属类簇的可能从属点。分配可能从属点时,若该未分配点的多个近邻被分配到同一个类簇,那么该未分配点也应该被分配到此类簇。

## 3　共享最近邻与客户交易数据聚类算法

依据上一章节中的聚类分配策略,给出了 PurTreeClust 局部密度和分离距离的计算方法,避免了 PurTreeClust 错误分配的问题。其基本思想:首先利用定义 4 和定义 5 计算局部密度和分离距离;然后从聚类中心出发,依据近邻信息先分配类簇的从属交易树,再分配类簇的可能从属交易树。Snn-PurTreeClust 聚类算法的伪代码,详见算法 1 ~ 算法 3。

在算法 1 中,首先计算交易树之间的 PurTree 距离,然后计算客户交易树的局部密度、分离距离和分离密度,通过分离密度筛选聚类中心。分配客户交易树时,依据交易树的近邻分配情况,利用算法 2 和算法 3 分配客户交易树。

在算法 2 中,首先将所有聚类中心压入队列,从聚类中心出发,判断该聚类中心的 $k$ 近邻是否满足公式 $|\mathrm{SNN}(i,j)|\geqslant k/2$,若满足则将该近邻交易树分配到该类簇,并将该近邻交易树压入队列。算法 2 通过聚类中心向外扩散,找到各聚类簇所有的从属交易树,并将其分配到对应的聚类簇,得到初步的聚类结果。

在算法 3 中,观察每一个未分配交易树的近邻分配情况,如果发现多个近邻被分配到同一个聚类簇中,那么该未分配交易树也有可能被分配到这个聚类簇。

首先建立一个矩阵 $\boldsymbol{M}$,矩阵 $\boldsymbol{M}$ 的行代表未分配交易树,列代表聚类簇。通过一次循环,找到矩阵 $\boldsymbol{M}$ 中的最大值,该最大值的行代表当前最需要被分配的交易树,列代表该未分配交易树的所属类簇,将其分配到该聚类簇中。如果一次循环后未找到最需要被分配的交易树,则增大近邻数 $k$,扩大搜索范围。

算法 1:Snn-PurTreeClust 聚类算法
输入:客户交易树集合 $Q$,近邻数 $k$,簇数 $m$
输出:聚类结果 $\Phi = \{C_1, C_2, \cdots, C_m\}$
算法开始
计算交易树之间的 PurTree 距离
计算交易树之间的共享近邻相似度;
for each $q \in Q$ do
计算 $q$ 的局部密度 den($q$);
计算 $q$ 的分离距离 sdis($q$);
计算 $q$ 的分离密度 sden($q$) = den($q$) * sdis($q$);
end for
筛选聚类中心集合 $U = \{q \in Q: \forall q' \notin U, \text{sden}(q) > \text{sden}(q')\}$,使得 $|U| = m$;
AssignSubTree($Q, U, k, m$);
AssignPossSubTree($Q, U, k, m$);
算法结束

算法 2:AssignSubTree(客户交易树集合 $Q$,聚类中心集合 $U$,近邻数 $k$,簇数 $m$)
输出:初步聚类结果 $\Phi = \{C_1, C_2, \cdots, C_m\}$
算法开始
初始化空队列 $P$,将所有聚类中心压入队列 $P$;
while $P$ 非空 do
弹出队列头元素 $p$;
for all $p$ 的邻居交易树 $n$ do
if $n$ 未被分配到任何簇且满足公式 $|\text{SNN}(p, n)| \geq k/2$ then
将 $n$ 分配到 $p$ 所在的簇;
将 $n$ 压入队列;
end if
end for
end while
算法结束

算法 3:AssignPossSubTree(客户交易树集合 $Q$,聚类中心集合 $U$,近邻数 $k$,簇数 $m$)
输出:最终聚类结果 $\Phi = \{C_1, C_2, \cdots, C_m\}$
算法开始
while 有交易树未被分配 do
建立分配矩阵 $\boldsymbol{M}$,矩阵行代表未分配交易树,矩阵列代表聚类簇;
for all 未分配交易树 $p$ do
for all $p$ 的邻居交易树 $q$ do
使矩阵行为 $q$,矩阵列为 $p$ 的值+1;
end for
end for
筛选矩阵 $M$ 中的最大值 max;
if max > 0 then
记录 max 所在的矩阵行 row 和矩阵列 col;
将第 row 个交易树分配到 col 聚类簇;
else
$k = k + 1$;
end if
end while
算法结束

假设有 $n$ 棵客户交易树,近邻数为 $k$,聚类簇数为 $m$。由上述算法描述可知,交易树之间的 PurTree 距离时间复杂度为 $O(n^2)$,共享近邻相似度、局部密度和分离距离的时间复杂度分别为 $O(kn^2)$、$O(kn)$ 和 $O(n^2)$,筛选聚类中心集合时间复杂度为 $O(n\log n)$,算法 2 与算法 3 的时间复杂度分别为 $O(mn^2)$ 和 $O((k+m)n^2)$,因此 Snn-PurTreeClust 算法总的时间复杂度为 $O((k+m)n^2)$。

## 4 实验结果分析

为验证 Snn-PurTreeClust(SPTC)算法的聚类效果,实验采用 6 个真实数据集对文中算法进行测试和评价,并与 PTC[14]、DBSCAN[17]、2 种谱聚类算法[18-19]和 3 种凝聚层次聚类算法[20]进行了对比分析。

在表 1 所示的 6 个真实交易数据集中,D1、D2、D3 是 3 个超市交易数据集,分别包含 795 个客户的 9 995 笔交易记录、795 个客户的 9 995 笔交易记录、1 179 个客户的 51 200 笔交易记录。D4、D5、D6 是从 kaggle 比赛一年的历史交易数据中构建的 3 个子集,该数据集一共包括 30 多万个客户的 3.49 亿笔交易记录。

表 1 6 个真实交易数据

| Data | Size | Customers | Attribute |
|---|---|---|---|
| D1 | 9 995 | 795 | 3 |
| D2 | 9 995 | 795 | 3 |
| D3 | 51 200 | 1 179 | 3 |
| D4 | 1 595 600 | 1 753 | 4 |
| D5 | 1 967 800 | 2 181 | 4 |
| D6 | 11 507 680 | 10 941 | 4 |

采取与 PurTreeClust 同样的方法确定最佳的聚类簇数,首先选定了 14 组簇数 $k$,在 D2 上运行 Snn-PurTreeClust 算法,并计算了间距统计量值[21],结果如图 2。可以看出,当 $\gamma = \{10, +\infty\}$ 时,Gap 值接近甚至小于 0,说明这两个参数无法找到簇类结构。当 $\gamma = \{0, 0.2, 0.5, 1\}$ 时,Gap 在 $k = 4$ 时陡然增加,因此为了

更好地揭示 D2 数据集的聚类结构,选用 $\gamma=0.5$, $k=4$ 且近邻数 $m=28$ 来进行下列部分实验。

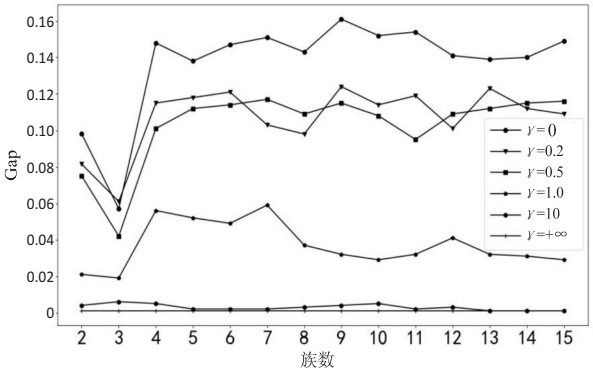
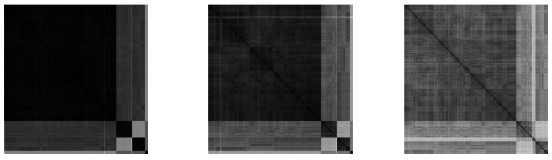
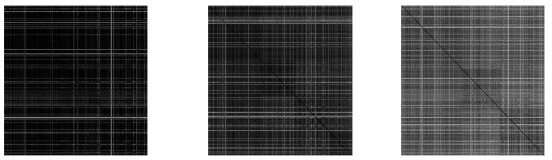



图 2　Snn-PurTreeClust 在 D2 上六种聚类结果的 Gap 值

为了更加直观地观测 Snn-PurTreeClust 算法的聚类效果,利用 PurTreeClust 算法中聚类结果的表示方式,将聚类结果绘制成图,如图 3 所示。将同一类簇中的交易树安置在一起,让其紧挨着,并且使行与列以相同的顺序表示交易树,图中深色表示较小的距离,浅色表示较大的距离。从图中可以清晰地观测到,当 $k=4$ 且 $\gamma=\{0.2,0.5,1\}$ 时,Snn-PurTreeClust 算法均可以发现更加紧凑清晰的类簇,PurTreeClust 算法均不能发现较紧凑清晰的类簇。经过上述对比,可以验证 Snn-PurTreeClust 算法在不同参数下均可以发现较紧凑清晰的类簇,具有更好的伸缩性,聚类效果比 PurTreeClust 算法更优秀。



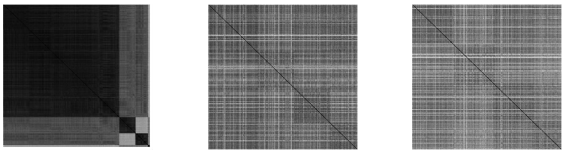
(a)SPTC( $\gamma$ =0.2, $k$ =4)　(b)SPTC( $\gamma$ =0.5, $k$ =4)　(c)SPTC( $\gamma$ =1, $k$ =4)



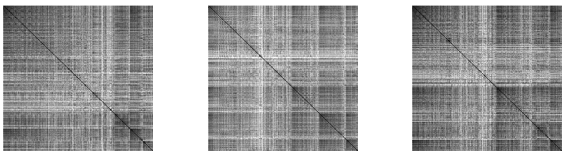
(d)PTC( $\gamma$ =0.2, $k$ =4)　(e)PTC( $\gamma$ =0.5, $k$ =4)　(f)PTC( $\gamma$ =1, $k$ =4)

图 3　PTC 与 SPTC 在 D2 上的聚类结果

为了进一步检验 Snn-PurTreeClust 算法的聚类效果,将 Snn-PurTreeClust 算法与六种聚类算法进行对比,所有算法均使用 PurTree 距离,其中参数 $\gamma=0.5$,簇数 $k=4$。六种聚类算法的结果如图 4 所示。从图中可以看出,DBSCAN 算法可以发现较清晰紧凑的类簇,其余五种算法均没有发现清晰紧凑的类簇,由此可说明,Snn-PurTreeClust 算法具有较好的聚类效果,可以准确发现比较清晰紧凑的类簇。



(a)DBSCAN 聚类结果　(b)NCut 聚类结果　(c)RCut 聚类结果



(d)HAC-S 聚类结果　(e)HAC-M 聚类结果　(f)HAC-C 聚类结果

图 4　六种聚类算法在 D2 上的聚类结果 ($\gamma=0.5, k=4$)

采用 6 个数据集来比较 Snn-PurTreeClust 算法与之前七种聚类算法的聚类效果。在本实验中,八种聚类算法均使用 PurTree 距离,其中参数 $\gamma$ 设置为 $\{0, 0.2, 0.5, 1, 10, +\infty\}$。选定同样 14 组 $k$ 值来运行除 DBSCAN 外的其他七种算法。对于每一种算法的聚类结果,分别计算簇内离散度 $\log(W(k))$,结果如表 2 所示。由表 2 可知,Snn-PurTreeClust 算法在 6 个数据集上的聚类结果均具有较低的簇内离散度 $\log(W(k))$,说明 Snn-PurTreeClust 可发现更紧凑的聚类簇,与其他七种聚类算法相比,Snn-PurTreeClust 算法的聚类效果更优异。

表 2　八种算法在 6 个数据集上的平均簇内离散度比较(粗体为最佳结果)

| Data | DAN | HAC-S | HAC-M | HAC-C | NCut | RCut | PTC | SNN-PTC |
|---|---|---|---|---|---|---|---|---|
| D1 | 4.27 | 4.29 | 4.26 | 4.24 | 4.28 | 4.29 | 4.22 | **4.21** |
| D2 | 4.26 | 4.33 | 4.29 | 4.31 | 4.33 | 4.36 | 4.27 | **4.25** |
| D3 | 4.64 | 4.69 | 4.65 | 4.65 | 4.67 | 4.68 | 4.62 | **4.59** |
| D4 | 5.18 | 5.19 | 5.13 | 5.14 | 5.15 | 5.16 | 5.12 | **5.11** |
| D5 | 5.40 | 5.44 | 5.35 | 5.33 | 5.43 | 5.46 | 5.35 | **5.32** |
| D6 | 7.46 | 7.44 | 7.43 | 7.43 | 7.44 | 7.45 | 7.42 | **7.39** |

## 5　结束语

利用客户交易数据聚类分析,可体现同簇客户拥有的相似消费习惯,从而获得了良好的客户细分效果。利用交易树之间的共享最近邻信息,该文提出一种客户交易数据聚类算法,可有效地发现更加紧凑清晰的类簇,避免了交易树错误分配,并通过 6 个客户交易数据集上的实验验证了该算法的有效性。未来如何降低

Snn-PurTreeClust 算法的时间代价有待进一步研究。

**参考文献:**


[1] 刘英姿,吴 昊. 客户细分方法研究综述[J]. 管理工程学报,2006,20(1):53-57.

[2] 刘 义,万迪昉,张 鹏. 基于购买行为的客户细分方法比较研究[J]. 管理科学,2003,16(1):69-71.

[3] FISHER R,DUBE L. Gender differences in responses to emotional advertising:a social desirability perspective[J]. Social Science Electronic Publishing,2005,31(4):850-858.

[4] 胡少东. 客户细分方法探析[J]. 工业技术经济,2005,24(7):66-69.

[5] 孙吉贵,刘 杰,赵连宇. 聚类算法研究[J]. 软件学报,2008,19(1):48-61.

[6] 孟增辉. 聚类算法研究[D]. 保定:河北大学,2005.

[7] 吴军英,辛 锐. 聚类分析在客户细分领域中的应用[J]. 微计算机信息,2010,26(28):199-200.

[8] KUO R J,HO L M,HU C M. Integration of self-organizing feature map and K-means algorithm for market segmentation [J]. Computers & Operations Research,2002,29(11):1475-1493.

[9] TSAI C Y,CHIU C C. A purchase-based market segmentation methodology[J]. Expert Systems with Applications,2004,27(2):265-276.

[10] LU T C,WU K Y. A transaction pattern analysis system based on neural network[J]. Expert Systems with Applications,2009,36(3):6091-6099.

[11] HSU F M, LU L P, LIN C M. Segmenting customers by transaction data with concept hierarchy[M]. [s. l. ]:Pergamon Press,Inc. ,2012:6221-6228.

[12] YU Z,LUO P,YOU J,et al. Incremental semi-supervised clustering ensemble for high dimensional data clustering[J]. IEEE Transactions on Knowledge & Data Engineering,2016,28(3):701-714.

[13] HOLY V,SOKOL O,CERNY M. Clustering retail products based on customer behaviour[J]. Applied Soft Computing,2017,60(3):32-39.

[14] CHEN X,FANG Y,YANG M,et al. PurTreeClust:a clustering algorithm for customer segmentation from massive customer transaction data[J]. IEEE Transactions on Knowledge & Data Engineering,2018,PP(99):132-138.

[15] MIGUEIS V L,CAMANHO A S,CUNHA J F E. Customer data mining for lifestyle segmentation[J]. Expert Systems with Applications,2012,39(10):9359-9366.

[16] LIU R,WANG H,YU X. Shared-nearest-neighbor-based clustering by fast search and find of density peaks[J]. Information Sciences,2018,450:200-226.

[17] SANDER J,ESTER M,KRIEGEL H P,et al. Density-based clustering in spatial databases:the algorithm GDBSCAN and its applications[J]. Data Mining & Knowledge Discovery,1998,2(2):169-194.

[18] HAGEN L,KAHNG A B. New spectral methods for ratio cut partitioning and clustering[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,2002,11(9):1074-1085.

[19] SHI J,MALIK J M. Normalized cuts and image segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2000,22(8):888-905.

[20] RUI X,WUNSCH D I. Survey of clustering algorithms[J]. IEEE Transactions on Neural Networks,2005,16(3):645-678.

[21] TIBSHIRANI R,HASTIE W T. Estimating the number of clusters in a data set via the gap statistic[J]. Journal of the Royal Statistical Society B,2001,63(2):411-423.