

文章编号:1001-9081(2015)07-1927-06

doi:10.11772/j.issn.1001-9081.2015.07.1927

## 基于近邻传播与密度相融合的进化数据流聚类算法

邢长征, 刘 剑\*

(辽宁工程技术大学 研究生院, 辽宁 兴城 125105)

(\*通信作者电子邮箱 954443316@qq.com)

**摘要:**针对目前数据流离群点不能很好地被处理、数据流聚类效率较低以及对数据流的动态变化不能实时检测等问题,提出一种基于近邻传播与密度相融合的进化数据流聚类算法(I-APDensStream)。此算法使用传统的两阶段处理模型,即在线与离线聚类两部分。不仅引进了能够体现数据流动态变化的微簇衰减密度以及在线动态维护微簇的删除机制,而且在对模型采用扩展的加权近邻传播(WAP)聚类进行模型重建时,还引进了异常点检测删除机制。通过在两种类型数据集上的实验结果表明,所提算法的聚类准确率基本能保持在95%以上,其纯度对比实验等其他相关测试都有较好结果,能够高实效、高质量、高效率地处理数据流数据聚类。

**关键词:**离群点; 数据流聚类; 近邻传播; 微簇

**中图分类号:** TP391; TP18    **文献标志码:**A

### Evolutionary data stream clustering algorithm based on integration of affinity propagation and density

XING Changzheng, LIU Jian\*

(Graduate School, Liaoning Technical University, Xingcheng Liaoning 125105, China)

**Abstract:** To solve the problems that the data stream outliers can not be disposed well, the efficiency of clustering data stream is low and the dynamic changes of data stream can not be real-time detected, an evolutionary data stream clustering algorithm based on integration of affinity propagation and density (I-APDensStream) was proposed. The traditional two-stage processing model was used in this algorithm, namely online and offline clustering. Not only the decay density of micro-cluster which could represent the dynamic changes of data stream and deletion mechanism for online dynamic maintenance of micro-cluster were introduced, but also the outliers' detection and simplification mechanism for model reconstruction by using the extended Weight Affinity Propagation (WAP) cluster was introduced. The experimental results on two types of data sets demonstrate that the cluster accuracy of the proposed algorithm remains at above 95%, and also achieves considerable improvements with respect to the purity compared to other algorithms. The proposed algorithm can cluster the data stream with high real-time, high quality and high efficiency.

**Key words:** outlier; data stream clustering; Affinity Propagation (AP); micro-cluster

## 0 引言

在当今快速发展的信息时代,各式各样的数据在人们的生活中随处可见,如何有效地从这些数据中得到人们想要的信息,一直是研究者们的研究热点。尤其对于那些快速持续到来,增长迅速,且复杂多变的数据流数据<sup>[1]</sup>,它们在网络、证券、医疗、电信等领域都有着不可限量的应用前景。然而传统的一些聚类算法已很难适应快速多变的数据流数据,国内外学者们对此研究了许多新的对于数据流挖掘的聚类算法,在此方面有了很大的进展。

Aggarwal等研究者设计了一种全新的流数据聚类架构CluStream<sup>[2]</sup>(stream clustering),首次运用两阶段处理方案:在线处理和离线阶段处理,其中第一部分在线算法主要用于统计分析数据流特征向量,得出其概要消息;而第二阶段则负责对应的用户请求,根据第一阶段所得信息产生最终比较准确的聚类结果。其最大的优点是它灵活的扩展性能;缺点则是该算法只是更符合于超球形布局的数据形态的聚类,而相对

于其他形态布局的流数据聚类则不是很理想。对此,Aggarwal等<sup>[3]</sup>研究者进一步设计了HPStream算法,其应用衰减函数和高维射影的方法来进行高维类型数据流的聚类分析,然仍需要提前设定维数均值;Cao等<sup>[4]</sup>设计出DenStream算法,在CluStream的两层架构基础上引进了潜藏的核心微簇以及孤立点微簇等概念,进而能够满足各种形状数据聚类。然而因为引进全局一致参数,所以聚类结果对参数值选取特别敏感;Chen等<sup>[5]</sup>设计的D-Stream算法,其引入了带衰减因子的密度计算法,能够实时调整数据流的进化特征,实现高效的密度网格聚类,但网格方法丢失了空间位置信息,处理边缘能力较差,因此结果将受网格粒度的影响较大;Chen等<sup>[6]</sup>进一步在D-Stream框架上引进了网格之间的吸引力的概念,对空间位置信息丢失问题进行了有效处理,但却使算法复杂度有所增加,运行效率降低。杨宁等<sup>[7]</sup>对于分布形态是倾斜数据类型的聚类效率不理想的问题,设计了局部密度阈值随时间自适应变化的目标关联覆盖算法,其缺点还是不能很好地解决任意形态分布的数据。接着Ntoutsi等<sup>[8]</sup>和于彦伟等<sup>[9]</sup>研究人

收稿日期:2015-01-15;修回日期:2015-03-25。    基金项目:国家自然科学基金资助项目(61402212)。

**作者简介:**邢长征(1967-),男,辽宁阜新人,教授,博士,主要研究方向:数据挖掘、数据库; 刘剑(1990-),男,湖南衡阳人,硕士研究生,主要研究方向:数据挖掘、数据库。

员分别运用了空间密度聚类和局部更新密度聚类进行数据流聚类,很好地解决了非球形数据聚类问题,存在的缺点是还不能较好地映射出数据流的演化信息。Zhang 等<sup>[10]</sup>在顶级会议 PKDD (Principles and Practice of Knowledge Discovery in Databases)第一次提到了 StrAP (Data streaming with Affinity Propagation) 算法,该算法思想是使新到达的各数据点与现有的数据模型进行匹配,假若匹配成功则进行现有模型的更新,不然将这些可能是异常点的数据点存入缓存盒中,但当检测到缓存盒中数据形态分布发生变化时,则将对现有整体模型进行重建。该算法能快速检测数据流的变化并能给出任意时段的聚类结果,但由于在缓存过程中,未能很好地处理异常值,而导致聚类效率不高。张建朋等<sup>[11]</sup>研究者进一步提出了基于密度与近邻传播(Affinity Propagation, AP)的数据流聚类算法 StrDenAP,此算法提出对微簇的动态删减机制,很好地处理历史微簇和现有微簇,使算法模型更加准确,但由于缺乏异常点删除机制,在模型重建时导致复杂度增高,其效率并没有较大提升。

经过上述一系列的算法研究,对于异常点的处理,聚类质量的提高等都没有较好的改善,因此,本文融合了近邻传播与密度的流聚类算法,设计了一种新算法,基于近邻传播与密度相融合的进化数据流聚类算法(Improved Data stream clustering algorithm based on integration of affinity propagation and density, I-APDenStream)。

本文算法仍引用经典的双层在线/离线处理模型,同时加入微簇衰减密度以及全新动态删除微簇机制的概念,以便更好地处理数据微簇的变化;而且在通过扩展的近邻传播聚类进行模型重建之前,本文加入异常点检测模式,更好地消除异常值,以便提高重建效率,最终得到更佳的聚类效果。经过后续实验可得出,新算法 I-APDenStream 在处理异常值、提高聚类精度方面更佳。

## 1 相关算法

### 1.1 近邻传播聚类

近邻传播(AP)是一种关于信息的近邻之间互传的聚类方法<sup>[12]</sup>。此算法主要通过利用数据点对相互之间的相似度作为基准,得到最佳的类代表点集,使得全部的数据点至它们相对应类代表点之间的相似度总和达至最大。算法依据  $n$  个数据点相互间的相似程度值构成相似度矩阵  $S$ ,然后根据此矩阵对角线上的值来判断点  $k$  是否可以作为类代表点,其标准是该数值若越大,则越有可能作为类代表点。其中,每个数据点和其余数据点间的相似程度相关概念如下。

1) 吸引度(responsibility)。 $R(i, k)$  定义为数据点  $i$  传至候选点  $k$  的消息,表示候选  $k$  点吸引  $i$  点的程度。

2) 归属度(availability)。 $A(i, k)$  定义为待选点  $k$  传至数据点  $i$  的消息,表示  $i$  点归属于  $k$  点的程度。

这两种类型连续不断地更新循环,只为找到更佳的类代表点,其过程即是近邻传播聚类算法的核心,其更新公式如下:

$$\begin{aligned} R(i, k) &= S(i, k) - \max\{A(i, j) + S(i, j)\}; \\ j &\in \{1, 2, \dots, N, j \neq k\} \\ A(i, k) &= \min\left\{0, R(k, k) + \sum_{i \neq k, j \neq k}^n \{\max[0, R(j, k)]\}\right\}; \\ i &\neq k, j \neq k \\ A(k, k) &= \sum_{j \neq k}^n \{\max(0, R(j, k))\}; i = k \end{aligned}$$

AP 算法通过上述公式不断循环更新信息,依据  $R(i, k)$  和  $A(i, k)$  的值大小来判决候选数据点  $k$  能否成为模型聚类中心,若值越大,则几率越高。AP 算法即是凭借在持续不断的的消息互传流程中频繁地刷新吸引度与归属度值,以致最后得到  $N$  个较佳的聚类代表点,然后把剩余的其他各数据点分别有效地分配到对应的各聚类中。

为了更好地适应数据流的特点,得出最佳的聚类结果,对近邻传播算法进行了稍加扩展,提出一种加权的 AP(Weighted AP, WAP)<sup>[13]</sup> 算法。其原理极类似于 AP 算法,唯一区别是权值算法不一样。对于 AP,其利用反复释义的数据项数目个数作为权值。而对于 WAP 的权值,则是把已聚类好的簇视为一个数据项来看待,同时用此簇的聚类中心作代表,则其权值即为此聚类簇内去除其聚类中心之后所剩余数据项的数量,因此在这些权值点上用 WAP 聚类,就相当于用 AP 对所有数据点的聚类。此加权算法在最不理想情况下时间复杂度为  $O(N^{3/2})$ ,即当数据量  $N$  很大时,WAP 算法的时间复杂度相对于 AP 算法  $O(N^2)$  明显降低,因而 WAP 算法更适合于大规模的数据流类型挖掘问题的解答。

### 1.2 DenStream 聚类

DenStream(Density Stream)聚类算法是在 CluStream 的两层架构原理上引进了潜在的密集微簇以及孤立点微簇等定义,以适应不同形状数据的聚类。此算法具体流程如下所示。

```
算法 1 DenStream 聚类。
预设一个密度阈值 u;
当微簇密度 < u, 则看作离群微簇; 否则看作潜在核心微簇;
While( 新到数据点  $x_i$  )
    首先将其归并于最近的潜在核心微簇
    If 不匹配
        将其放入最近的离群类簇
        If 成功
            将此离群类簇密度  $u_1$  与  $u$  相比较
            If(  $u_1 > u$  )
                将此类簇转为核心类簇
            Else
                继续接收数据, 重复上述步骤
            Else
                新建一个离群微簇
            Else
                直接纳入最近核心微簇
        End;
    End;
```

DenStream 算法有效地处理了噪声数据,而且对于任意形态的数据均有较好的效果。但由于设定全局一致阈值,使得结果对参数的选择特别依赖。对此,黄德才等<sup>[14]</sup>采用了滑动窗口技术对它进行了稍微改进,设计了改进算法 I-DenStream(Improved Density Stream),能够动态地更新数据以及调整阈值。

## 2 I-APDenStream 算法模型与相关定义

### 2.1 算法模型

I-APDenStream 算法采用三过程处理法,即初始模型、在线模型以及离线处理模型。其中初始模型即是运用最开始的数据来初始化接下来的在线处理模型。而在线处理部分则是进行动态更新聚类微簇,以及时刻监测新到数据流的分布变化情况等,得出在线部分聚类结果。最后离线处理则是根据在线聚类结果,由用户调用,并根据用户的查询得出最终的挖

掘结果。此模型利用在线/离线两部分协调处理以满足动态变化、快速的数据流数据,能够很好地服务于用户以及他们对流数据挖掘分析的需求。算法模型如图1所示。

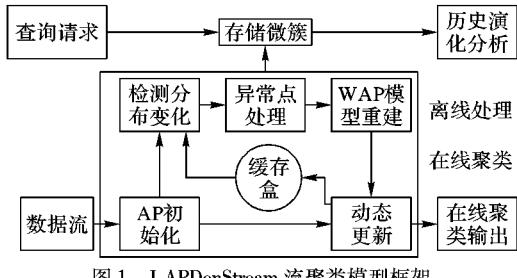


图1 I-APDenStream 流聚类模型框架

## 2.2 基本定义与扩展

**定义1** 衰退函数。数据流数据速度快,数量大且动态变化持续不断,然而对于数据的挖掘研究需要时效性,因此随着时间越长,对于相对较久的历史数据,其重要性也将越来越低,并将其用一个衰退函数来表示:  $f(t) = 2^{-\lambda t}$ ,  $\lambda \in (0, 1)$ 。此函数的值表示数据的重要程度。

**定义2** 元组密度函数。设数据流中任意时刻  $t_0$  所到数据点为  $x$ , 则定义在  $t$  时刻的元组密度函数  $f_x(x, t)$  为:  $f_x(x, t) = 2^{-\lambda(t-t_0)}$ ,  $\lambda \in (0, 1)$ 。

**定义3** 微簇密度函数。设微簇  $mc$  在  $t$  时刻的构成组合为  $n$  个相互近邻的元组, 则其密度函数定义为:  $f_{mc}(t) = \sum_{i=1}^n f_x(x_i, t)$ 。

**定理1** 微簇  $mc$  在数据流中的总和最大值必低于  $1/(1 - 2^{-\lambda})$ , 因此其均值也必将低于  $1/[n(1 - 2^{-\lambda})]$ , 其中  $n$  为当前微簇数量。

**证明** 由前面定义1可得知, 数据点的重要性随时间而变化, 且以函数  $f(t) = 2^{-\lambda t}$  衰退, 又由定义2、定义3的密度函数计算公式可得:

$$f_{mc}(t) = \sum_{i=1}^n f_x(x_i, t) = \sum_{i=0}^{t_m} 2^{-\lambda t}$$

其中,  $t_m (t_m \rightarrow \infty)$  定义为当前时间, 则微簇的密度最大值为:

$$f_{mc}(t)_{\max} = \lim_{t_m \rightarrow \infty} \frac{1 - 2^{-\lambda(t_m+1)}}{1 - 2^{-\lambda}} = \frac{1}{1 - 2^{-\lambda}}$$

由此可得, 微簇总和将不会高于  $1/(1 - 2^{-\lambda})$ , 且当微簇数为  $n$  时, 其均值也不会高于  $1/[n(1 - 2^{-\lambda})]$ 。证毕。

**定义4** 核心微簇和稀疏微簇。定义  $t$  时刻的微簇密度为  $f_{mc}(t)$ , 则设制特定的阈值  $u$ (大于1), 若  $f_{mc}(t) \geq u/[n(1 - 2^{-\lambda})] = L$ , 则定义此微簇为核心微簇; 反之为稀疏微簇。

**定理2** 假设  $x$  为某一时刻  $t_1$  所接收到的新数据, 并进行微簇更新, 则在  $t_2 (t_1 < t_2)$  时刻接收新数据时的微簇动态更新公式为:

$$f_{mc}(mc, t_2) = 2^{-\lambda(t_2-t_1)} f_{mc}(mc, t_1) + 1$$

**证明** 由微簇密度公式可得在  $t_1$  时刻的微簇密度为:

$$f_{mc}(t_1) = \sum_{i=1}^n f_x(x_i, t_1), \text{ 则由元组密度函数有: } f_x(t_2) = 2^{-\lambda(t_2-t_1)} f_x(t_1), \text{ 则当在 } t_2 (t_1 < t_2) \text{ 吸收一个新的数据点时: } f_{mc}(mc, t_2) = f_x(t_2) + 1 = 2^{-\lambda(t_2-t_1)} f_{mc}(mc, t_1) + 1. \text{ 证毕。}$$

此定理说明微簇可以在吸收新数据时进行动态更新, 而不用时刻存储所到数据的时间信息, 进而能减少存储空间并提升聚类效率。

## 3 I-APDenStream 算法的具体实施

### 3.1 聚类初始化

把先前到来的一些数据子集利用AP算法将其执行有效聚类并求得最开始的StrAP模型过程定义为模型的初始化<sup>[15]</sup>。将最初的数据子集定义为聚类微簇, 若微簇数据密度超过核心微簇所定义的阈值  $L$ , 则以微簇内数据点及邻近点创建核心微簇, 并将全部核心微簇的半径均值定为最开始的  $R$ 。

### 3.2 在线微簇更新

在数据流聚类过程中, 数据流数据以快速、不断变化的形式涌进, 以致新的类簇频繁冒出, 而对于时间较长的旧簇则渐渐衰退。新出现的类簇需要慢慢积累达到密度阈值进而转为核心微簇, 因此绝大部分的新数据点都归于现有的稀疏微簇。所到新样本点将其划入最近的核心微簇(优先)或者稀疏微簇内, 若不能与现有模型所匹配, 则将其暂放入缓存盒内, 直至盒内有新的模式被挖出, 将其重建并融合于现有模型之中。根据算法1可得其算法如下。

#### 算法2 在线微簇更新

```

While( 新到样本点  $x_i$  )
    For ( 所有核心微簇 hc/ 稀疏微簇 xc )
        If (  $d(x_i, hc/xc) \leq R$  )
            把  $x_i$  归入与其最近的核心微簇 hc/ 稀疏微簇 xc 内;
            If (  $f_{mc}(t) \geq L$  )
                说明该稀疏微簇已转变成核心微簇, 可将其归入核心
                微簇类;
            Else
                继续吸收新数据直至转变;
            Else
                数据点  $x_i$  和现有微簇不匹配, 将其归入缓存盒内, 也由此
                表示此点可能是离群异常点或是新类簇种子;
        End
    
```

由于数据流数据快速且无限, 久而久之, 聚类微簇数量也必将快速增长以致占用大量内存, 为此, 本文加入了微簇处理机制。

#### 3.2.1 核心微簇的衰退检验

随着时间的增长, 稀疏微簇可以慢慢转为核心微簇。同样, 现有的核心微簇若长久不能吸收新的数据点, 其  $hc$  密度也会逐渐降低, 当低于阈值  $L$  时,  $hc$  将会返退为  $xc$ 。对此, 有必要时常检验核心微簇密度。然而, 任意核心微簇返退为稀疏微簇的最短跨度为:  $T_k = [\frac{1}{\lambda} \ln(\frac{L}{L-1})]$ , 此公式由  $2^{-\lambda T_k} L + 1 = L$  所得, 因此, 对微簇密度检验时间段应为  $T_k$ 。

#### 3.2.2 微簇的动态删减

对于快速增长的微簇数量, 若不及时处理, 将会很大程度加重内存负载。根据3.2.1节得知微簇是随着时间动态变化的, 稀疏微簇可以转变为核心微簇; 反之亦然。但是其过程有长有短, 有些稀疏微簇需要很长时间才能转变或许很难转变, 也有些核心微簇转为稀疏微簇后也很难再恢复到核心微簇, 对此, 本文引进了最低权值限度理论: 若任意稀疏微簇密度长期低于预定的阈值下限, 则可判断此微簇很难发展成核心微簇, 很有可能是异常点所组成, 可对其进行删除。其权值下限函数为:  $Q(T, T_0) = (2^{-\lambda(T-T_0+T_k)} - 1)/(2^{-\lambda T_k} - 1)$ 。由公式可知在微簇刚形成的时候, 即  $T = T_0$  有:  $Q(T, T_0) = 1$ 。而当  $T \rightarrow$

∞ 有:  $\lim_{T \rightarrow \infty} Q(T) = L$ 。由此说明,稀疏微簇的期望权值随时间增加。

由 3.1.1 和 3.1.2 节可得微簇处理机制算法如下。

### 算法 3 微簇的动态删减。

```

If (  $T \bmod T_K$  )
  For( 所有核心微簇 hc )
    If ( $f_{hc}(t) < L$ )
      该核心微簇已转变为离群微簇,将其删除;
    For( 所有稀疏微簇 xc )
      If ( $f_{xc}(t) < Q(T, T_0)$ )
        该稀疏微簇很难转为核心微簇,将其删除;
  End

```

### 3.2.3 缓存盒内异常点处理

由算法 2 可知,当数据点与当前模型不匹配时,则将其放入缓存盒内。此点可能是新簇的种子也可能是离群噪声点,当缓存盒内数据达到最大或者出现新的模式时,需对其进行模型重建。然而当初的离群噪声点以及其中可能的突变点并未及时处理,因此将会导致重建效率较低而且准确度不高,因此,本文使用暂存盒中的数据进行数据统计分析,得出其能反映数据的参数特性,根据此特性,删除异常值。具体过程如下:

- 1) 从接收到的数据中获取存储统计值(最大值  $Max$ 、最小值  $Min$ 、求和值  $Sum$  与平均值  $Avg$ )。
- 2) 扫描暂存盒并计算数据的标准偏差。
- 3) 设置采样参数  $\&$ ,  $\&$  能反映在该暂存盒中的数据分布。在后续实验中,本文选择在暂存盒中的流数据的标准偏差作为  $\&$ 。
- 4) 删除异常值。本文根据置信区间  $[Avg - \&, Avg + \&]$ ,保留区间内的数据,而除去区间外的数据。

具体算法如下。

### 算法 4 缓存盒内异常点处理。

```

新到达数据点  $x_t$ ;
For(缓存盒中所有数据点)
  计算  $Max, Min, Sum, Avg$ ;
  For(所得值)
    计算数据的标准偏差,并将其作为反映该缓存盒中数据分布的采样参数  $\&$ ;
    If (  $Avg - \& < x_t < Avg + \&$ )
      保留,进行下一步的重建;
    Else
      此点异于其他绝大多数数据点,删除此点,以便更好地进行
      下一步的重建;
  End

```

### 3.2.4 模型重建

当缓存盒中的数据渐渐增加时,有可能会出现新的数据类模型,因此,有必要将它们考虑到模型中去。除此之外,当缓存盒中的数据渐满达到最大时,同样需对其进行模型重建。模型重建是将当前模型的各类代表点与缓存盒中的各数据点经过重新聚类后得到一种新模型的过程。不仅考虑流数据的变化,而且能够符合大规模数据流聚类模式。

经过加权近邻传播聚类算法将模型重建后,缓存盒需要进行清空处理,以便接收新的数据。如此循环,可以得出新的聚类代表点、新的模型微簇向量以及概要信息,根据新的信息进行离线阶段的处理,依据用户的需求进行分析,得出最终的聚类结果。

### 3.3 离线分析处理

为更好地进行离线阶段处理,首先给出以下定义。

**定义 5 核心微簇相接。**若有两个核心微簇  $hc_i\{E_i, f_{hc_i}, \sum i, Max_i, t_n\}$  与  $hc_j\{E_j, f_{hc_j}, \sum j, Max_j, t_n\}$ , 其中:  $E$  为开始代表点,  $f$  为微簇密度,  $\sum j$  表示聚类代表点到所属元组的加权距离和,  $Max$  为取最远边界坐标点的坐标向量。服从如下条件。

1)  $1 - p \leq (\sum i_{mc_i}) / (\sum j_{mc_j}) \leq 1 + p$ ,  $p$  是预设的阈值。

2) 服从如下公式,即密度可达条件:

$$dist(e_i, e_j) \leq dist(Max_i, e_i) + dist(Max_j, e_j)$$

则认为  $hc_i$  与  $hc_j$  是体积形态相近,位置相邻的两个相接核心微簇。

离线处理过程的重点在于用户,依据用户的各种不一样的需求而得到相应的挖掘效果。用户可以利用任意时刻的快照,应用深度优先遍历的方法找出全部密度相接的核心微簇,得到更加精确的聚类微簇。也能够依据当前时间  $t$  与用户所要求的时间范围  $h$ ,找出  $t$  时刻与  $t - h$  时刻的快照,并将它们相减即可得出  $h$  时间范围内的微簇集差,因此可得到 3 个集合:{新出现的核心微簇元组}、{衰退删除的核心微簇元组}以及{存留的核心微簇元组},借此可分析得到数据流的演化结果。

### 3.4 I-APDenStream 算法整体描述

- 1) 模型初始化。应用近邻传播聚类算法得到初始聚类模型以及核心微簇。
- 2) 刷新现有模型。执行算法 2。
- 3) 微簇衰退与转变。核心微簇与稀疏微簇会随着时间的增长而互相转变。
- 4) 微簇动态维护。执行算法 3。
- 5) 异常点处理。执行算法 4。
- 6) 模型重建。当缓存盒满或者有新的模式出现时,需对其进行模型重建。
- 7) 转回步骤 2) 循环处理直至结束。
- 8) 离线分析处理。

## 4 实验分析

### 4.1 实验环境

实验平台 Intel Pentium Dual-Core CPU、主频 2.3 GHz、2 GB 内存、512 MB 独立显卡、Windows 7 操作系统,算法采用 Matlab 7.0 编写。

实验流程中所用到的数据是由 KDD Cup 99 数据集和森林数据集 Forestcove Type 构成。其中 KDD Cup 99 数据集源于美国麻省理工学院林肯研究室持续两周的网络流监测,分别蕴含于 23 种不一样的网络连接类别,本文在 42 个有效属性之间选取 34 个持续性属性以组成实验数据,从中取 1% 的样本数据集进行实验。而对于森林数据集 Forestcove Type 则是源于美国森林服务消息系统,任意一组数据都带有 54 维属性,且其中带有 10 维连续性属性,总共 581 010 组数据记录。全部实验数据都已经过处理至  $[0, 1]$  区间内。

实验对 I-APDenStream 聚类效果进行分析评价,且与 StrAP 算法与传统的 DenStream 以及其改进算法 I-DenStream 进行相对比较。实验过程中所需参数预设如下: 初始化过程数据流数据个数  $N_0 = 2000$ , 微簇衰退因子  $\lambda = 0.25$ , 初始微簇半径均值为  $R$ , 阈值参数  $u = 3$ , 剩余参数与传统算法中设置一样,且以  $v$  为数据流动速度。

#### 4.2 I-APDenStream 聚类质量分析

首先对算法 I-APDenStream 进行聚类准确度评价分析, 分别在两种数据集上在相同实验参数下, 对 I-APDenStream 与 StrAP 以及 DenStream、I-DenStream 进行对比分析, 得到图 2。由图 2 可知, 实验开始的时候聚类准确率都有所降低, 是因为当数据点越来越多时, 没有及时对其进行处理并删除异常值。而 I-APDenStream 算法在不同数据集都能够一直保持较高准确率, 说明异常值的删除非常有利于提高聚类的准确率。对于 StrAP 算法等其他算法在数据量达到很大时, 准确率有所上升, 说明异常点毕竟是非常少的一类数据, 当数据量相当大的时候, 比开始的时候准确率会有所上升, 因此, I-APDenStream 算法比 StrAP 等算法具有更高的准确度。

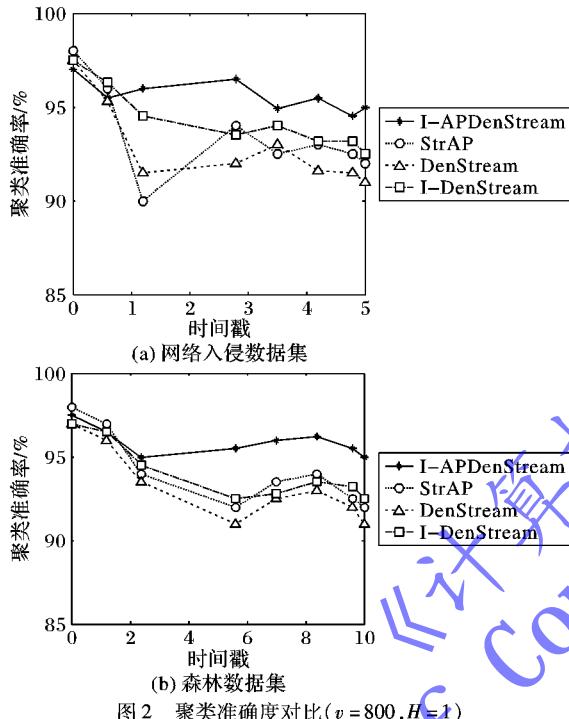


图 2 聚类准确度对比 ( $v=800, H=1$ )

除了对准确度的分析与对比, 本文还将 I-APDenStream 算法在这两种数据集上分别与 StrAP 算法、经典的 DenStream 及其改进算法 I-DenStream 实行聚类纯度的互比。为得到更佳的结果, 算法分别执行 5 次计算聚类纯度均值作为最后结果。从图 3 中可以看出在流速  $v=800, H=1$  的情况下, I-APDenStream 算法相比其余算法关于 KDD Cup 99 数据流聚类纯度最好, 然后是 StrAP 和 I-DenStream 算法, 其中滑动窗口  $\Delta$  的大小对 StrAP 算法有些许影响, 而对于 I-DenStream 算法因在原有基础上作了改进, 所以聚类纯度要高于原有算法。但还是低于本文算法, 因此说明本文算法更优于其他算法。在森林数据集上聚类效果也可看出本文算法聚类纯度一直处于最高水平。

从以上准确度和纯度对比实验可得出, 本文算法具有较高聚类质量。

#### 4.3 执行时间实验

对于数据流聚类算法, 所需时间的长短也是检验算法的好坏的重要因素, 因此, 本文将 I-APDenStream 算法、StrAP 算法以及 DenStream 的改进算法在两种数据集上进行实验对比。从图 4 可看出, 3 种算法的运行时间都随着数据流量的增加而增长, 但 I-APDenStream 算法的增长速度明显低于其余两种。这是由于 I-APDenStream 算法可以动态维护微簇以

及处理分析好异常值, 从而提高模型重建效率, 大幅提升整个聚类效率; 而 DenStream 的改进算法开始略慢于近邻传播流聚类算法, 后面由于部分改进而处理速度提高, 但仍低于本文新算法, 故而可得知本文算法执行速度快、效率高。

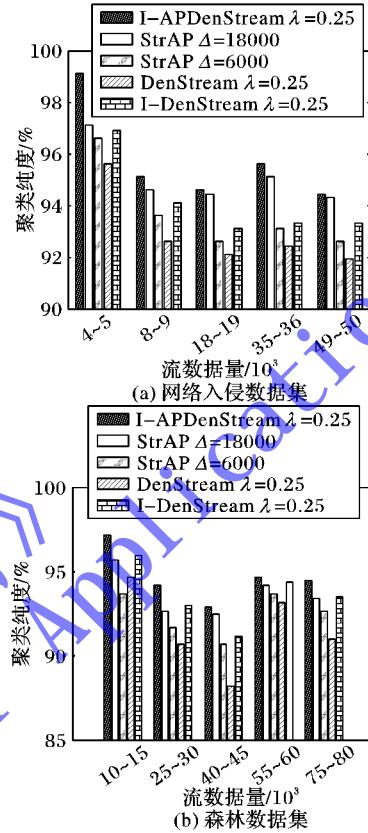


图 3 聚类纯度对比 ( $v=800, H=1$ )

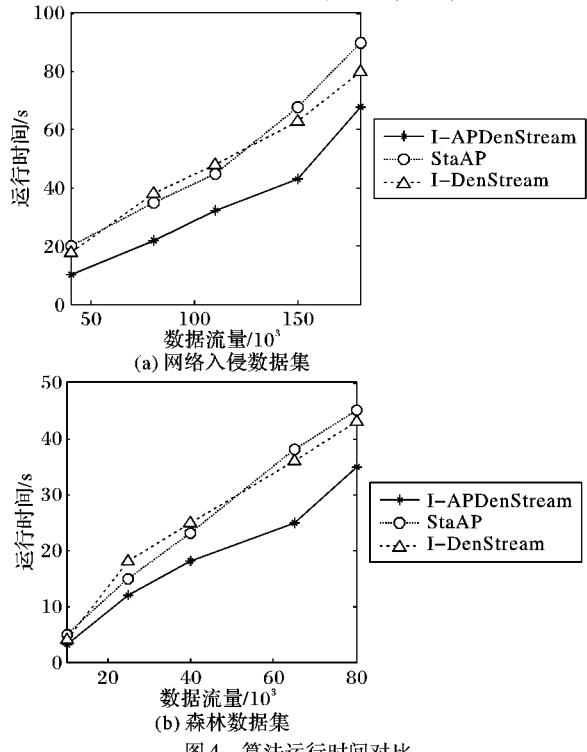


图 4 算法运行时间对比

#### 4.4 稳定性分析

为了更好地检验此新算法, 在复杂度不同以及聚类簇数不同的数据流上对 I-APDenStream 的执行时间进行比较, 以

体现随着聚类簇数增多或数据复杂度增大的情况下的稳定性。从图 5(a)可看出 I-APDenStream 算法聚类数据流的运行时间和簇数的关系是线性增长关系,而且表现得比较平稳,时间增长幅度不大,这说明簇数随着各数据的增多并未大幅增加(其中: $B$ 表示数据量)。从图 5(b)也可看出,随着数据复杂度的增大,其执行时间也随之增长,但变化幅度也比较平稳。综上可得出,I-APDenStream 算法在数据量过多且较复杂的情况下,稳定性也保持较高水平。

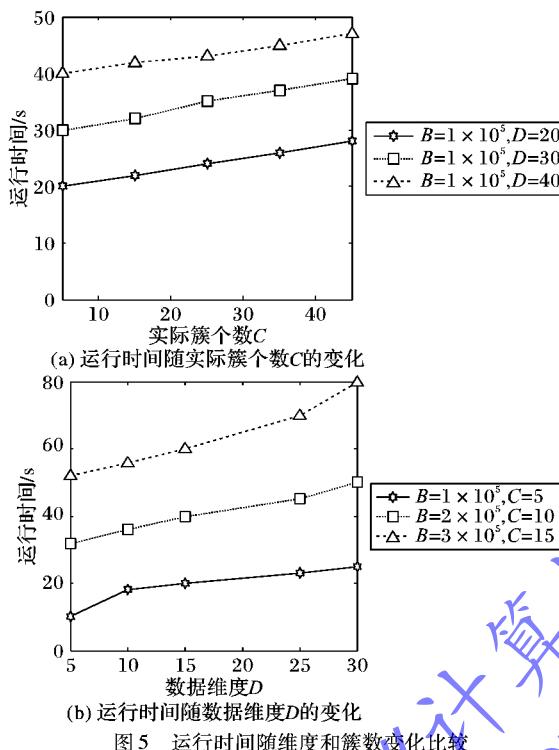


图 5 运行时间随维度和簇数变化比较

#### 4.5 参数敏感性测试

本文新算法执行过程中预设了两个主要参数: $\lambda$ (微簇衰退因子)和 $u$ (微簇离群点因子)。参数设置的好与坏对算法最终聚类结果有一定的影响,所以很有必要对其进行有效分析。本文算法设置的 $\lambda = 0.25$ ,因此按此数展开扩大它的取值范围( $0.025 < \lambda < 2.5$ ),经过实验得出随 $\lambda$ 变化,其聚类纯度的变化如图 6 所示。从图 6 可看出 $\lambda$ 过低或过高都不利于聚类结果,若想聚类结果持续稳定且质量纯度较高,则应尽量控制此因子在 $0.1 \sim 0.8$ 。而对于微簇离群点阈值 $u$ ,虽然有最低限制,但同样不能过高,因为预设得太高将会影响稀疏微簇向核心微簇的转变,以致于长时间为转变即把它当作离群微簇而删除,从而减少了核心微簇的数量,降低了聚类质量。从图 6 可看出,应尽量将 $u$ 控制在 $1 \sim 7$ 。

## 5 结语

本文为了得到更佳的数据流聚类结果,设计了一种新的聚类算法——I-APDenStream。此算法是将近邻传播流聚类算法与密度流聚类算法相互融合,并加入微簇维护机制与异常点检测删除机制,同时在更新过程通过加权的近邻传播进行模型重建。通过与传统类似算法的一些性能比较,此新算法具有更高的聚类质量、聚类效率以及稳定性等优势性能,在实际过程中适用性高,能得到理想的聚类结果。然而,由于此算法需预设部分参数,因此不能完全自适应在线聚类,而此问题也是将要进一步研究的方向。

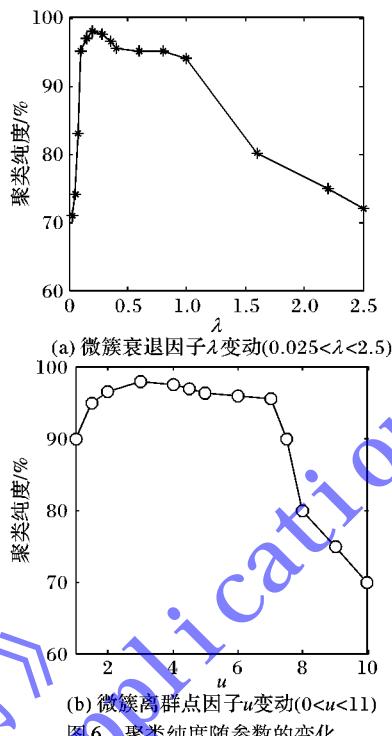


图 6 聚类纯度随参数的变化

#### 参考文献:

- [1] HAN J, KAMBER M, PEI J. Data mining: concepts and techniques [M]. FAN M, MENG X, translated. 3rd ed. Beijing: China Machine Press, 2013: 288 - 347. (汉江, 姬伯明, 裴健. 数据挖掘: 概念与技术[M]. 范明, 孟小峰, 译. 3 版. 北京: 机械工业出版社, 2013: 288 - 347.)
- [2] AGGARWAL C C, HAN J, WANG J, et al. A framework for clustering evolving data streams [C]// Proceedings of the 29th International Conference on Very Large Data Bases. [S. l.]: VLDB Endowment, 2003: 81 - 92.
- [3] AGGARWAL C C, HAN J, WANG J, et al. A framework for projected clustering of high dimensional data streams [C]// Proceedings of the 30th International Conference on Very Large Data Bases. [S. l.]: VLDB Endowment, 2004: 852 - 863.
- [4] CAO F, ESTER M, QIAN W, et al. Density-based clustering over an evolving data stream with noise [C]// Proceedings of the Sixth SIAM International Conference on Data Mining. Philadelphia: SIAM, 2006: 328 - 339.
- [5] CHEN Y, TU L. Density-based clustering for real-time stream data [C]// Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2007: 133 - 142.
- [6] CHEN Y, TU L. Stream data clustering based on grid density and attraction [J]. ACM Transactions on Knowledge Discovery from Data, 2009, 3(3): 12 - 20.
- [7] YANG N, TANG C, WANG Y, et al. Clustering algorithm on data stream with skew distribution based on temporal density [J]. Journal of Software, 2010, 21(5): 1031 - 1041. (杨宁, 唐常杰, 王悦, 等. 一种基于时态密度的倾斜分布数据流聚类算法. 软件学报, 2010, 21(5): 1031 - 1041.)
- [8] NTOUTSI I, ZIMEK A, PALPANAS T, et al. Density-based projected clustering over high dimensional data streams [C]// Proceedings of the Twelfth SIAM International Conference on Data Mining. Philadelphia: SIAM, 2012: 987 - 998.

(下转第 1949 页)

3)词边界统计指标知识库的获取需要消耗大量计算资源。

4)统计指标阈值的范围需要根据经验人为设定。

本文的识别结果有待进一步提高,下一步可考虑增加实体结构方面规则;也可考虑设计以词为标注单位的条件随机场模型,以便将本文中用于判断实体边界的统计指标直接融入模型,完全由统计模型决定实体边界。

#### 参考文献:

- [1] LAFFERTY J, McCALLUM A, PEREIRA F. Conditional random fields: probabilistic models for segmenting and labeling sequence data [C]// ICML 2001: Proceedings of the 2001 International Conference on Machine Learning. San Francisco: Morgan Kaufmann Publishers, 2001: 282 – 289.
- [2] FENG Y, SUN L, ZHANG D, et al. Study on the Chinese named entity recognition using small scale tail hints [J]. *Acta Electronica Sinica*, 2008, 36(9): 1833 – 1838. (冯元勇,孙乐,张大鲲,等.基于小规模尾字特征的中文命名实体识别研究[J].电子学报,2008,36(9):1833 – 1838.)
- [3] XIANG X, SHI X, ZENG H. Chinese named entity recognition system using statistics-based and rules-based method [J]. *Journal of Computer Applications*, 2005, 25(10): 2404 – 2406. (向晓雯,史晓东,曾华琳.一个统计与规则相结合的中文命名实体识别系统[J].计算机应用,2005,25(10):2404 – 2406.)
- [4] SHE J, ZHANG X. Musical named entity recognition method [J]. *Journal of Computer Applications*, 2010, 30(11): 2928 – 2931. (余俊,张学清.音乐命名实体识别方法[J].计算机应用,2010,30(11):2928 – 2931.)
- [5] ZHANG J, WANG S, QIAN C. CRF and rules-based recognition of medical institutions name in Chinese [J]. *Computer Applications and Software*, 2014, 31(3): 159 – 162. (张金龙,王石,钱存发.基于CRF和规则的中文医疗机构名称识别[J].计算机应用与软件,2014,31(3):159 – 162.)
- [6] LIU F, ZHAO J, LYU B, et al. Study on product named entity recognition for business information extraction [J]. *Journal of Chinese Information Processing*, 2006, 20(1): 7 – 13. (刘非凡,赵军,吕碧波,等.面向商务信息抽取的产品命名实体识别研究[J].中文信息学报,2006,20(1):7 – 13.)
- [7] ZHANG C, GUO J, XIAN Y, et al. Named entity recognition of the products with English based on conditional random fields [J]. *Computer Engineering and Science*, 2010, 32(6): 115 – 117. (张朝胜,郭剑毅,线岩团,等.基于条件随机场的英文产品命名实体识别[J].计算机工程与科学,2010,32(6):115 – 117.)
- [8] MEI F. Research on product named entity recognition and normalization [D]. Harbin: Harbin Institute of Technology, 2011. (梅丰.产品名实体识别及规范化研究[D].哈尔滨:哈尔滨工业大学,2011.)
- [9] LUO F, XIONG Q, XIAO M. Product named entity recognition based on ontology [J]. *Journal of Wuhan University of Technology: Information and Management Engineering*, 2011, 33(6): 948 – 952. (罗芳,熊前兴,肖敏.基于本体的产品命名实体识别研究[J].武汉理工大学学报:信息与管理工程版,2011,33(6):948 – 952.)
- [10] LI H. Statistical learning methods [M]. Beijing: Tsinghua University Press, 2012: 195 – 197. (李航.统计学习方法[M].北京:清华大学出版社,2012:195 – 197.)
- [11] CHEN F, LIU Y, WEI C, et al. Open domain new word detection using conditional random field method [J]. *Journal of Software*, 2013, 24(5): 1051 – 1060. (陈飞,刘奕群,魏超,等.基于条件随机场方法的开放领域新词发现[J].软件学报,2013,24(5): 1051 – 1060.)
- [12] ZHAO X, ZHANG H. New words identification based on iterative algorithm [J]. *Computer Engineering*, 2014, 40(7): 154 – 158. (赵小宝,张华平.基于迭代算法的新词识别[J].计算机工程,2014,40(7):154 – 158.)
- [13] FENG Y, SUN L, ZHANG D, et al. A rapid algorithm to Chinese named entity recognition based on single character hints [J]. *Journal of Chinese Information Processing*, 2008, 22(1): 104 – 110. (冯元勇,孙乐,李文波,等.基于单字提示特征的中文命名实体识别快速算法[J].中文信息学报,2008,22(1):104 – 110.)
- [14] KUDO T. CRF++: yet another CRF toolkit [EB/OL]. [2014-12-01]. <http://crfpp.googlecode.com/svn/trunk/doc/index.html>.
- [15] LIU D, NOCEDAL J. On the limited memory BFGS method for large scale optimization [EB/OL]. [2014-12-07]. <http://users.iems.northwestern.edu/~nocedal/PDFfiles/limited-memory.pdf>.

(上接第1932页)

- [9] YU Y, WANG Q, KUANG J, et al. An on-line density-based clustering algorithm for spatial data stream [J]. *Acta Automatica Sinica*, 2012, 38(6): 1051 – 1058. (于彦伟,王沁,邝俊,等.一种基于密度的空间数据流在线聚类算法[J].自动化学报,2012,38(6): 1051 – 1058.)
- [10] ZHANG X, FURTLEHNER C, SEBAG M. Data streaming with affinity propagation [C]// Proceedings of the 2008 Machine Learning and Knowledge Discovery in Databases, LNCS 5212. Berlin: Springer, 2008: 628 – 643.
- [11] ZHANG J, CHEN F, LI S, et al. Data stream clustering algorithm based on density and affinity propagation techniques [J]. *Acta Automatica Sinica*, 2014, 40(2): 277 – 288. (张建朋,陈福才,李绍梅,等.基于密度与近邻传播的数据流聚类算法[J].自动化学报,2014,40(2):277 – 288.)
- [12] LIU D, JIANG M. Affinity propagation clustering on oral conversa-

- tion texts [C]// ICSP 2012: Proceedings of 2012 IEEE 11th International Conference on Signal Processing. Piscataway: IEEE, 2012: 2279 – 2282.
- [13] ZHANG X, FURTLEHNER C, GERMAIN-RENAUD C, et al. Data stream clustering with affinity propagation [J]. *IEEE Transactions on Knowledge and Data Engineering*, 2014, 26(7): 1644 – 1656.
- [14] HUANG D, WU T. Density-based clustering algorithm for mixture data sets [J]. *Control and Decision*, 2010, 25(3): 416 – 421. (黄德才,吴天虹.基于密度的混合属性数据流聚类算法[J].控制与决策,2010,25(3):416 – 421.)
- [15] FUJIWARA Y, IRIE G, KITAHARA T, et al. Fast algorithm for affinity propagation [C]// Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence. Piscataway: IEEE, 2011: 2238 – 2243.