ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.9, September 2010, pp.2173−2182
doi: 10.3724/SP.J.1001.2010.03654
© by *Institute of Software*, *the Chinese Academy of Sciences*. All rights reserved.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种不确定数据流聚类算法*

张 晨[1], 金澈清[2+], 周傲英[2]

[1](复旦大学 计算机科学技术学院 上海市智能信息处理重点实验室,上海 200433)

[2](华东师范大学 软件学院 上海市高可信计算重点实验室,上海 200062)

Clustering Algorithm over Uncertain Data Streams

ZHANG Chen[1], JIN Che-Qing[2+], ZHOU Ao-Ying[2]

[1](Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai 200433, China)

[2](Shanghai Key Laboratory of Trustworthy Computing, Software Engineering Institute, East China Normal University, Shanghai 200062, China)

+ Corresponding author: E-mail: cqjin@sei.ecnu.edu.cn

Zhang C, Jin CQ, Zhou AY. Clustering algorithm over uncertain data streams. *Journal of Software*, 2010, 21(9):2173−2182. http://www.jos.org.cn/1000-9825/3654.htm

Abstract: This paper proposes a novel algorithm, named EMicro, to cluster uncertain data streams. Although most of the works used today mainly use the distance metric to describe the cluster quality, EMicro considers distance metric and data uncertainty together to measure the clustering quality. Another contribution of this paper is the outlier processing mechanism. Two buffers are maintained to reserve normal micro-clusters and potential outlier micro-clusters, respectively, to obtain good performance. Experimental results show that EMicro outperforms existing methods in efficiency and effectiveness.

Key words: uncertain data stream; clustering; outlier

摘 要: 提出了 EMicro 算法,以解决不确定数据流上的聚类问题.与现有技术大多仅考虑元组间的距离不同,EMicro 算法综合考虑了元组之间的距离与元组自身不确定性这两个因素,同时定义新标准来描述聚类结果质量.还提出了离群点处理机制,系统同时维护两个缓冲区,分别存放正常的微簇与潜在的离群点微簇,以期得到理想的性能.实验结果表明,与现有工作相比,EMicro 的效率更高,且效果良好.

关键词: 不确定数据流;聚类;离群点

中图法分类号: TP391 **文献标识码**: A

近年来,随着信息技术不断发展,数据流模型在许多应用中广泛出现,其特征是数据到达速度极快、规模庞大,例如 Internet 应用、传感器网络等[1].此外,受物理仪器精度限制、周围环境等因素的影响,流数据还同时具有

* Supported by the National Natural Science Foundation of China under Grant Nos.60933001, 60803020 (国家自然科学基金); the National Science Foundation for Distinguished Young Scholars of China under Grant No.60925008 (国家杰出青年基金项目); the Shanghai Leading Academic Discipline of China under Grant No.B412 (上海市重点学科建设项目)

Received 2008-11-17; Revised 2009-02-24; Accepted 2009-04-29

© 中国科学院软件研究所 http://www.jos.org.cn

不确定性.因此,面向不确定数据流的分析与挖掘技术已成为新近的研究热点.本文研究聚类(clustering)问题,即如何将数据集合划分为若干个簇(cluster),并保证簇内元组的相似性高,簇间元组的相似性低.

现有数据流聚类算法大多针对确定性数据流,各元组均为确定性元组.但是,由于这些算法仅考虑元组间的距离因素,并不考虑元组的存在概率等不确定因素,因而无法直接应用到不确定数据流中去.以图 1 为例,各元组按照相互间的距离被划分成 A,B,C 和 D 这 4 个簇.在簇 B 与簇 D 内,各元组的空间位置关系完全相同,距离平方和(sum of square distance,简称 SSQ)也相同.进一步观察会发现:簇 B 中元组的存在概率较高,在 70%~90%之间;而簇 D 中的元组的存在概率则小得多,在 10%~30%之间.传统聚类算法无法描述这种差异,迫切需要针对不确定数据流的特点提出新的解决方案.

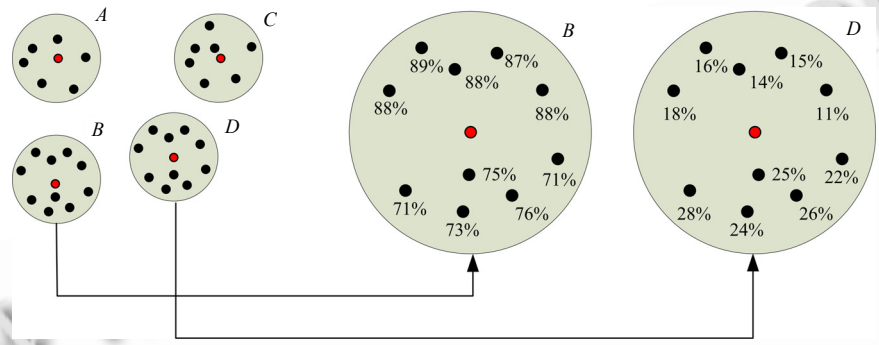


Fig.1 Impacts of data uncertainty for clustering results

图 1 不确定性对聚类结果的影响

离群点检测与处理是与数据流聚类相关的一个重要问题,不同算法所采用的策略不同.在 CluStream 算法[2]中,若新元组无法被现有微簇(micro-cluster)所吸收,则首先合并两个靠得最近的微簇,然后再为新元组创建一个微簇.例如在图 2 中,初始阶段存在 A,B,C,D 这 4 个簇;在 T_1 时刻,元组 X_1 到达,导致 C,D 两个簇合并,且另外为 X_1 创建一个新微簇.易知,若元组 X_1 是一个离群点,则该合并操作会降低聚类质量.UMicro 算法[3]的策略有所不同.例如,假设 T_2 时刻来了新的元组 X_2,UMicro 会删除一个最久未更新的微簇,并为元组 X_2 创建一个新微簇.易知,若 X_2 也是一个离群点,该删除操作亦会降低聚类质量.

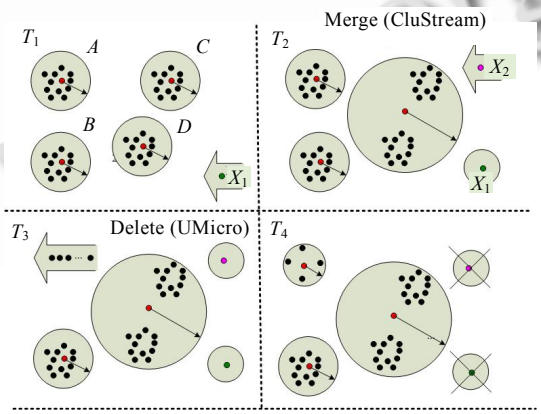


Fig.2 CluStream vs. UMicro

图 2 CluStream 算法与 UMicro 算法的对比

本文提出了一种针对不确定数据流的聚类算法 EMicro,主要贡献包括:1) 充分考虑存在级不确定性对聚类问题的影响,提出了新的数据结构(UCF)来描述微簇,并定义了新标准来描述聚类质量;重新定义了不确定簇

© 中国科学院软件研究所 http://www.jos.org.cn

的聚类特征与质量标准,使聚类过程能够兼顾距离与不确定性双重因素;2) EMicro 算法的一个关键步骤是新元组的被吸附准则,本文在制定该准则时兼顾了距离因素和元组的不确定因素;3) 设计了新机制来应对离群点情况.最后,基于真实数据集和模拟数据集的实验表明,EMicro 具有良好的聚类质量、较快的处理速度,能够有效适应不确定数据流场景.

本文第 1 节介绍相关工作.第 2 节定义不确定数据流模型,提出新的数据结构以描述微簇.第 3 节详细描述 EMicro 算法的各个步骤,分析离群点处理机制.第 4 节提供实验结果及其分析.第 5 节对全文做总结并指出后续研究方向.

1 相关工作

1.1 数据流聚类

数据流聚类问题得到了广泛的研究.STREAM 算法首先将数据流划分成若干个小段,在各个小段内部分别应用传统的聚类方法(如 *k*-means 或 *k*-medians)进行处理,最终将这些中间数据整合成聚类结果[4].CluStream 算法采用在线维护微簇和离线高层聚类的框架,用金字塔框架(pyramid framework)来存储微簇快照[2].文献[5]提出了一种采用空间分割、组合以及按密度聚类的算法 ACluStream.Babcock 等人利用指数直方图 EH[6]来维护滑动窗口内相关簇的统计信息[7].D-Stream 算法则采用基于密度的数据流聚类方法,充分运用剪枝技术,空间复杂度低,且能观测各个微簇的演化情况[8].

1.2 不确定数据聚类

典型的面向不确定数据的聚类算法包括 FDBSCAN[9],FOPTICS[10]和 UK-means[11].这些算法主要针对静态不确定数据库,而非不确定数据流.FDBSCAN 算法是基于密度的 DBSCAN 算法的改进版本,元组之间的距离被定义为模糊距离(fuzzy distance),需要通过两个元组的概率密度函数计算得到.FOPTICS 算法是 OPTICS 算法的改进版本[10].UK-means 算法改进了传统的 *k*-means 算法,利用最小边界矩形(MBR)描述数据点可能出现的区域,并通过设计剪枝策略来降低计算复杂度.

1.3 不确定数据流聚类

Aggarwal 等人提出了 UMicro 算法,以针对不确定数据流进行聚类[3].他们将 CF(clustering feature)结构[12]扩展成 ECF 结构,增加了描述不确定性的部分,从而计算各元组与簇之间的距离以及簇半径.

近期,Aggarwal 为了解决高维数据聚类所面临的“维度灾难”问题,提出了一种投影空间下的不确定数据流聚类算法[13].同样,为了突破 UMicro 只能用于某些特定不确定数据模型的限制,文献[14]提出一种基于信息熵的不确定性数据流聚类算法,采用信息熵来衡量元组的不确定性信息,并在聚类过程中综合考虑不确定性与距离双重因素的影响.

上述工作均仅考虑了元组的属性级不确定性,并未考虑元组的存在级不确定性.本文所提出的 EMicro 算法则主要面向含存在级不确定性的不确定数据流.

2 不确定数据流模型

元组的不确定性可以通过多种方式进行描述,本文研究点概率模型(point probability case).在该模型中,元组的属性值确定,而存在性不确定,用一个[0,1]之间的概率值表示[15−17].点概率模型应用广泛,例如在 RFID 应用中,RFID 读卡器存在漏读、多读等现象,所读数据的存在性并不确定.一个不确定数据流是一个由不确定元组构成的序列.

定义 3.1(不确定数据流). 不确定数据流 *S*(或称点概率流)是一个由相互独立的 *k* 维不确定性元组构成的序列, $S=\{(\overline{X_1},p_1),(\overline{X_2},p_2),...,(\overline{X_n},p_n)\}$,其中, $\overline{X_i}$ 是第 *i* 个元组的值,p_i 是该元组的存在概率,$0\leqslant p_i\leqslant 1$.

文献[3]中定义的 ECF 结构能够概括由不确定数据组成的簇,并可计算中心点、半径等重要统计信息.但是

© 中国科学院软件研究所 http://www.jos.org.cn

ECF结构仅能处理属性级不确定性,无法处理存在级不确定性,因而无法针对点概率流进行挖掘.鉴于此,本文提出了UCF结构以对点概率流进行概括,同时也能够计算一些重要统计量.

定义 3.2(UCF). 由一组具有时标 $T_1,\ldots,T_n$ 的 k 维元组 $\overrightarrow{X_1},\ldots,\overrightarrow{X_n}$ 所组成的不确定聚类特征(uncertain clustering feature,简称UCF)可描述为 $2k+3$ 维的向量 $(\overrightarrow{PF2},\overrightarrow{PF1},n,p_c,t)$.其中: $\overrightarrow{PF2}$ 为各元组的概率加权平方和,包含 k 项,第 m 项的值为 $\overrightarrow{PF2}^{(m)}=\sum_{i=1}^{n}p_i(\overrightarrow{X_i}^{(m)})^2$; $\overrightarrow{PF1}$ 为各元组的概率加权线性和,也包含 k 项,第 m 项的值为 $\overrightarrow{PF1}^{(m)}=\sum_{i=1}^{n}p_i\overrightarrow{X_i}^{(m)}$; n 为簇包含的元组个数; p_c 为所有元组的概率和 $p_c=\sum_{i=1}^{n}p_i$; t 为集合中最新元组的时标 $t=\max(T_i)$.

性质 3.1. 令 U 是针对簇 C 的UCF,则当 C 中新加入元组($\overrightarrow{X}$,p)后,U 也可相应更新.

该性质的正确性非常直观.当新元组加入之后,字段 t 被更新为新的时间戳,字段 n 的值加1,其余字段的值均线性增加.

在现有的确定性聚类算法中,簇的中心点是簇内元组的几何中心.即在任一维度,中心点的值是各元组的平均值.但是,几何中心无法反映不确定元组的概率分布情况.例如在图1中,尽管簇 B 和簇 D 的几何中心一致,但是簇 B 中概率较高的点主要分布于上方,而簇 D 中概率较高的点主要分布于下方.显然,如果以概率值来描述元组的权重,则簇 B 的中心点应在其几何中心稍上位置,而簇 D 的中心点应在其几何中心稍下位置.因此,有必要重新定义簇的中心点与半径,使之适应不确定数据模型.

定义 3.3. 簇的概率中心 C_p 定义为簇内元组的概率加权线性均值,即 $C_p=\overrightarrow{PF1}/p_c$.

容易验证,若各元组的存在概率均为100%时,概率中心与几何中心匹配.

半径是簇的重要统计数据,描述簇的规模以及对新元组的吸附能力.在现有工作中,半径一般被定义为簇内各元组到中心点的距离平方和的均值的开方根.现以加权想法重新定义包含不确定元组的簇的半径,概率高的元组对簇的半径值影响更为显著,如下所示:

$$R^2=\left(\sum_{i=1}^{n}p_i\left(\overrightarrow{X_i}-\frac{\overrightarrow{PF1}}{p_c}\right)^2\right)\Big/p_c=\left(\sum_{i=1}^{n}\left(p_i\overrightarrow{X_i}^2-2\frac{p_i\overrightarrow{X_i}\overrightarrow{PF1}}{p_c}+\frac{p_i\overrightarrow{PF1}^2}{p_c^2}\right)\right)\Big/p_c=\frac{\overrightarrow{PF2}}{p_c}-\frac{\overrightarrow{PF1}^2}{p_c^2}.$$

由图1可以观察到,簇的质量不仅与簇内元组的相似性程度有关,也与簇内元组的存在概率有关.一般来说,如果簇内元组的相似性越高(即半径越小),则簇的质量越高;如果簇内元组的平均存在概率越高,则簇的质量也会提升.鉴于此,我们定义 Q 来描述簇的质量.

定义 3.4. 一个簇的质量 Q 与簇内元组的平均概率成正比,与半径成反比,即 $Q=p_c/(n\cdot R)$.

对于任一簇而言,新元组的加入会导致簇的质量 Q 出现3种可能的变化情况:首先,若该元组的概率值高于簇的平均概率值时,且该元组与中心点距离较近,则该元组的加入能够提高簇的质量;其次,若新元组的概率值低于簇的平均概率值,且该元组离中心点距离较远,则该元组的加入将降低簇的质量;最后,若新元组概率高且离中心点远,或者概率低且离中心点近,则簇质量的变化趋势需要通过进一步的计算才能得到.

3 聚类算法

本节描述新的EMicro算法,以处理不确定数据流聚类问题.如图2所示,在现有的主流数据流聚类算法(例如CluStream或UMicro)中,当新元组无法被任何现有微簇吸收时,则将其认定为种子点,并创建新微簇.然而,若新元组实际上是离群点而非新簇的起点时,这些策略却对现有微簇集合产生了负面的影响.EMicro算法拟采用一种新缓冲机制来处理这种情况.

首先,在内存中保存两类缓冲区:核心微簇缓冲区(BUF_C)和离群点微簇缓冲区(BUF_O),分别存放微簇(即 UCF).核心微簇缓冲区对正常元组进行聚类;离群点缓冲区用于检验离群点,并判断是否将其提升到核心微簇缓冲区之中或者直接删除.令 n_c 与 n_o 分别表示核心微簇缓冲区和离群点缓冲区的规模,n_{ratio} 表示算法使用的最大微簇数目,易知 $n_{ratio}=n_c+n_o$.EMicro算法的细节见算法1.

© 中国科学院软件研究所 http://www.jos.org.cn

算法 1. Emicro

1. 将核心微簇缓冲区 BUF_C 与离群点微簇缓冲区 BUF_O 初始化为空;
2. **while** (新元组($\overline{X}$,p)到达)
3. **if** (核心微簇缓冲区未满,即$|BUF_C|<n_c$)
4. 创建仅包含($\overline{X}$,p)的 *UCF*,并加入 BUF_C 中;
5. **else**
6. C_c=*FindOptimalCluster*($\overline{X}$,p,BUF_C);
7. **if** (C_c is not *NULL*) //说明($\overline{X}$,p)能够被某一核心微簇所吸收
8. 核心微簇 C_c 吸收($\overline{X}$,p);
9. **else**
10. **if** (离群点微簇缓冲区未满,即$|BUF_O|<n_o$)
11. 创建仅包含($\overline{X}$,p)的 *UCF*,并加入 BUF_O 中;
12. **else**
13. C_o=*FindOptimalCluster*($\overline{X}$,p,BUF_O);
14. **if** (C_o is not *NULL*) //说明($\overline{X}$,p)能被某一离群点微簇所吸收
15. 离群点 C_o 吸收($\overline{X}$,p);
16. **else**
17. delete ($\overline{X}$,p); //($\overline{X}$,p)是全局离群点
18. *CheckClustersProcess*(); //对微簇进行调整维护

在 EMicro 中,对于新到达的元组($\overline{X}$,p),首先判断核心微簇缓冲区是否已满,如果未满,则创建仅包含($\overline{X}$,p)的 *UCF*,并加入到 BUF_C 中(第 4 行);反之,若核心微簇缓冲区已满,则需通过调用 FindOptimalCluster 函数来判断这个新元组是否能够被现有的核心微簇缓冲区成员所吸收.若返回结果不为 *NULL*,则 C_c 将吸收($\overline{X}$,p)(第 8 行);若返回 *NULL*,则认定该新元组是一个潜在的离群点,需做进一步的判断.如果离群点微簇缓冲区未满,则创建一个仅包含($\overline{X}$,p)的微簇,加入到 BUF_O 中去(第 11 行);否则,尝试在离群点微簇缓冲区中寻找最合适的微簇吸收之.如果找不到目标簇,则直接删除该元组(第 13~17 行).最后,算法调用 CheckClustersProcess 函数对微簇缓冲区进行调整维护(第 18 行).

3.1 簇选择算法FindOptimalCluster

FindOptimalCluster 用于在一个微簇集合中找到一个能够吸收元组($\overline{X}$,p)的最合适的微簇.首先,计算各个微簇的半径以及元组($\overline{X}$,p)与各微簇的概率中心之间的距离.若元组到簇的距离大于簇半径的τ倍(通常τ=3[2]),则可认为该元组无法被任何簇所吸收,返回 *NULL*;反之,该元组能够被某一簇吸收.若存在多个满足半径限制的微簇,则返回一个最优的微簇(即概率引力值最高,见定义 4.1).

定义 4.1(概率引力). 元组($\overline{X}$,p)到微簇 $UCF=(\overline{PF2},\overline{PF1},n,p_c,t)$ 的概率引力为 p_c/d^2,其中,d 为元组($\overline{X}$,p)到 *UCF* 的概率中心的距离.

定义 4.1 源于一个朴素的观察:当新元组($\overline{X}$,p)越靠近一个微簇,则越有可能被该微簇所吸附;当微簇内元组的概率和越大时,则该微簇内的元组越密集,对新元组($\overline{X}$,p)的吸附能力越强.事实上,这个思想与物理学中的万有引力思想有相似之处.

3.2 簇进化算法CheckClustersProcess

算法 2.

1. 核心微簇缓冲区与离群点微簇缓冲区分别以λ_C和λ_O衰减;
2. **while** (两个缓冲区存在某微簇,其权重比低于ρ)
3. 从缓冲区中移除该微簇;
4. **while** ($\min(w(u)|u\in BUF_C)<\max(w(u)|u\in BUF_O)$)

© 中国科学院软件研究所 http://www.jos.org.cn

5. 在两个缓冲区间交换相应的微簇;
6. **while** (核心微簇缓冲区 BUF_C 未满,且离群点微簇缓冲区 BUF_O 非空)
7. 将 BUF_O 中权重值等于 $\max(w(u)|u\in BUF_O)$的微簇移到 BUF_C 中;

在 CheckClustersProcess 算法中,首先需要对两个缓冲区进行衰减操作.所谓衰减,是指元组(或微簇)的重要性随着时间的推移而逐步减弱,新元组的重要性比旧元组高.常用的衰减方法是指数衰减法.若元组($\overline{X}$,p)在 t_1 时刻到达,则在 t_2 时刻,该元组的权重为 $w(\overline{X},p)=p\cdot 2^{-\lambda(t_2-t_1)}$,其中,$\lambda$为衰减速率.$\lambda$的取值越大,则衰减越快.微簇的权重也可以依此定义.假设微簇 u 包含 n 个元组$\{(x_i,p_i)\}$,分别于 $T_1,\ldots,T_n$ 到达,则在 T 时刻微簇 u 的权重为 $w(u)=\sum_{i=1}^{n}p_i 2^{-\lambda(T-T_i)}$.易知,每过一个单位时间,微簇的权重等于在原权重基础上乘以 $2^{-\lambda}$.如果为每个 UCF 额外附加表征权重的 w 字段,则该字段总是容易维护的.CheckClustersProcess 算法对核心微簇缓冲区与离群点微簇缓冲区采用不同的衰减速率,分别记为λ_C 和λ_O.如果数据流中存在较大的进化性,希望聚类过程能够敏感地反映出这种进化特征时(即尽快地发现新簇),那么应该设置$\lambda_O<\lambda_C$;反之,如果用户更加关注结果的稳定性,或者根据领域知识能够判断出数据流进化的相对平缓(不会经常出现新簇)的情况下,那么可以设置$\lambda_C<\lambda_O$(第 1 行).

其次,检查各个缓冲区是否存在过于陈旧的微簇,若有,则从内存中移除.我们采用权重比参数进行检验.如前所述,微簇的 p_c 字段表示该微簇的概率和,则可将微簇 u 的权重比定义为 $w(u)/p_c$,该值位于[0,1]之间.然后可将权重比与预定义的参数ρ进行比较,若小于ρ,则表示构成该微簇的元组大多较为陈旧,可以将其从内存中移除(第 2~3 行).

再次,随着新元组的不断到达,两个缓冲区不断演化,导致核心微簇缓冲区中的部分微簇的权重反而低于离群点微簇缓冲区中的微簇.此时就触发交换机制,将离群点微簇缓冲区中权重最高的微簇移到核心微簇缓冲区中,将核心微簇缓冲区中权重最低的微簇移到离群点微簇缓冲区中,最终使得核心微簇缓冲区中的微簇均比离群点微簇缓冲区中的微簇更加重要(第 4~5 行).

最后,若核心缓冲区未满,则需要从离群点缓冲区中将权重最高的微簇移至核心微簇缓冲区中,直至核心微簇缓冲区充满为止(第 6~7 行).

3.3 分 析

CheckClustersProcess 算法提供了两个参数来实现对离群点的支持:第 1 个参数是缓冲比参数,描述离群点微簇缓冲区的大小占全部缓冲区大小的比率,记为$\alpha=n_o/(n_c+n_o)$.假设总空间开销不变,则缓冲比的值越高,可分配给潜在的离群点的空间越大,对离群点的支持越充分;第 2 个参数是衰减速率比($\lambda_{ratio}=\lambda_c/\lambda_o$),描述不同缓冲区的衰减速度比较.换句话说,该参数描述了对不同类型的微簇的倾向性,可以通过对核心微簇与缓冲微簇采用不同的衰减率来调整是倾向于重视离群点还是倾向于忽视离群点.

4 实验分析

4.1 实验设置

本节以 UMicro[3]为基准算法来验证 EMicro 的性能.两个算法均由 matlabV14R7 工具实现,运行在一台配置了 Pentium IV 3.0GHz 的 CPU 的电脑上,操作系统是 Windows XP Professional.

我们首先设计 3 个确定性数据集合,包括一个模拟数据集和两个真实数据集,然后再将其转化成相应的不确定数据集.确定性数据集如下所示:

- 模拟数据集 Syndrift.首先,随机选定一系列簇中心点,半径也随机设定.随着时间推移,不断调整半径值(每次调整$\pm\varepsilon$),以模拟簇漂移现象.簇内元组的总体分布满足高斯分布,总共产生 60 万条记录.
- 真实数据集 1:网络入侵检测(network intrusion detection)数据集.该数据集包括一个局域网内某一时段的 TCP(transfer control protocol)连接日志记录,含持续时间、传输字节数等属性.每条记录对应于一个正常的连接或者是 4 种类型的网络攻击之一,例如拒绝服务攻击、非授权访问远程机器攻击等.我们从 42 个属性中选取 34

© 中国科学院软件研究所 http://www.jos.org.cn

个数值型属性.

- 真实数据集 2:Forest Covertype 数据集[18],含 581 012 个元组,从全部 54 个属性中选取 10 个数值型属性.

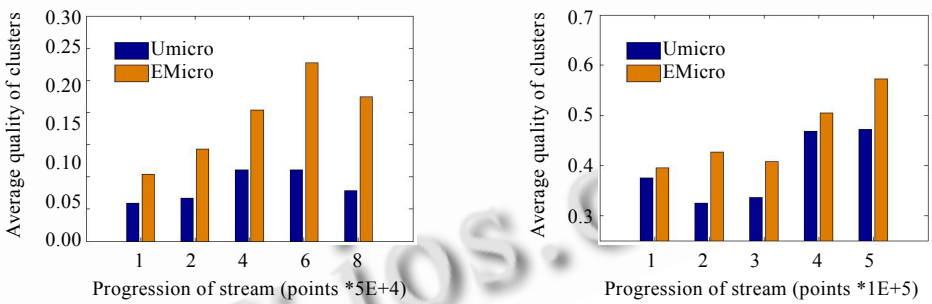
UMicro 与 EMicro 所针对的不确定数据流模型不同.EMicro 算法仅考虑存在级不确定性,而不考虑属性级不确定性;UMicro 算法仅考虑属性级不确定性,而不考虑存在级不确定性,二者无法运行在同一不确定性数据集合上.因此,本文采取一种折衷的方法:即从同一确定性数据集合 D 开始分别构造不确定数据集 D_1 和 D_2,D_1 仅含存在级不确定性,D_2 仅含属性级不确定性.换句话说,对于任一 D 中的元组 X,转变到 D_1 之中的元组时描述为(X,p),p 在[0,1]之间随机选取;转变到 D_2 之中的元组时,元组的期望值仍为 X,方差是$(1-p)X$.易知,若 p 值较大,则在 D_1 和 D_2 中对应元组的不确定性均较小;反之若 p 值较小,则在 D_1 和 D_2 中对应元组的不确定性均较大.同时,拟在将来工作中考虑如何基于同一数据集进一步分析两种算法的性能优劣.

现有方法采用距离平方和 SSQ 描述聚类结果的质量,但是该度量仅考虑距离因素,无法描述存在概率对聚类结果的影响.因此,本文采用平均簇质量 AQ 作为度量基准,即 $AQ=avg(Q_i)$.

EMicro 与 UMicro 采用相同的离线聚类算法,详见文献[2,3,19].其基本思想是:把任意核心微簇 C_i 看作是一个落在其概率中心且权重为 $w(C_i)$的虚拟点,再采用加权 k-means 算法对这些虚拟点进行聚类,以生成聚类结果.除特别标明,EMicro 算法中的参数设置如下:缓冲比 α=3%,衰减速率比 λ_{ratio}=1,收纳半径阈值 τ=3.UMicro 算法的参数设置为:权重比阈值 ρ=1/50,误差参数 μ=0.5,衰减系数 λ=0.1,收纳半径阈值 τ=3.

4.2 聚类效果

图 3 分别比较了 EMicro 和 UMicro 在网络入侵检测数据集和森林数据集上的聚类结果质量.横轴是数据流量,纵轴是平均质量.可以看出,EMicro 的性能指标远优于 UMicro,其主要原因是,EMicro 在分配新元组到现有微簇的过程中同时考虑了距离因素和概率因素.尽管该方法并非以最小化 *SSQ* 为目标,但在很多情况下,EMicro 的 *SSQ* 指标也会优于 UMicro,见表 1.原因是 UMicro 算法引入了误差维度,导致单个元组的可能范围变大,从而增大了 *SSQ* 计算过程中的簇内距离,进而降低了聚类质量.



(a) Network intrusion detection dataset
(a) 网络入侵检测数据集

(b) Forest covertype dataset
(b) 森林覆盖类型数据集

Fig.3 Quality comparison between EMicro and UMicro

图 3 EMicro 与 UMicro 的聚类质量比较

Table 1 *SSQ* comparison between EMicro and UMicro

表 1 EMicro 与 UMicro 的 *SSQ* 的比较

(a) Network intrusion detection dataset					
	0.5×10^5	1×10^5	2×10^5	3×10^5	4×10^5
UMicro-SSQ	2.95E+11	1.38E+11	1.38E+11	7.19E+10	9.18E+10
EMicro-SSQ	1.69E+11	9.65E+10	8.19E+10	9.97E+10	2.42E+10

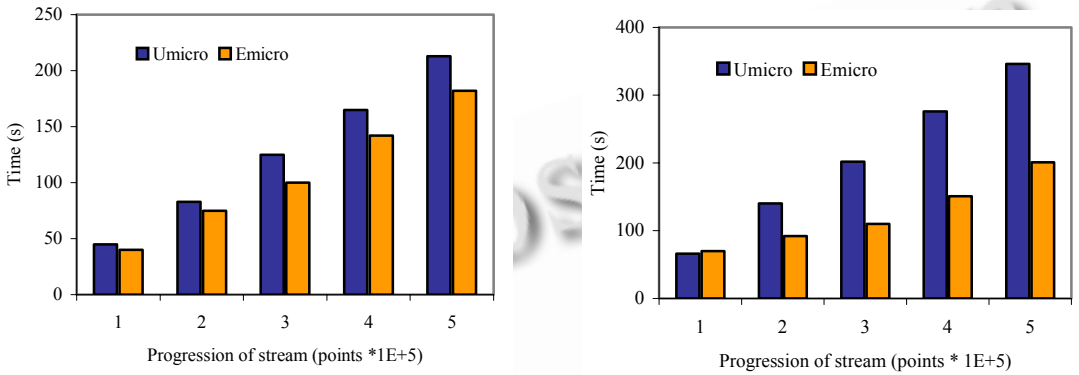
(b) Forest covertype dataset					
	1.0×10^5	2×10^5	3×10^5	4×10^5	5×10^5
UMicro-SSQ	5.21E+09	6.07E+09	2.7543+09	3.76E+09	2.54E+09
EMicro-SSQ	3.91E+09	1.31E+09	5.97E+08	3.30E+08	1.19E+07

© 中国科学院软件研究所 http://www.jos.org.cn

4.3 时间复杂度

时间复杂度是数据流算法的重要指标,影响 EMicro 与 UMicro 时间开销的主要部分为在线统计微簇信息和存储快照的频率.为了比较两个算法的效率,我们设置相同的快照频率,然后比较在线微聚类部分的性能.

图 4 展示了 EMicro 在不同数据集上的处理结果,横轴表示数据流量,纵轴表示处理时间. UMicro 与确定数据相比,由于需要增加一套维度来描述误差信息,因此会加大每个维度上的计算开销;同时,维度较低带来的开销下降大于计算概率因素带来的性能消耗.实验结果显示,EMicro 与 UMicro 算法的执行时间都会随数据流长度呈线性增长,可以达到每秒数千点的处理速度,但 EMicro 算法更快. EMicro



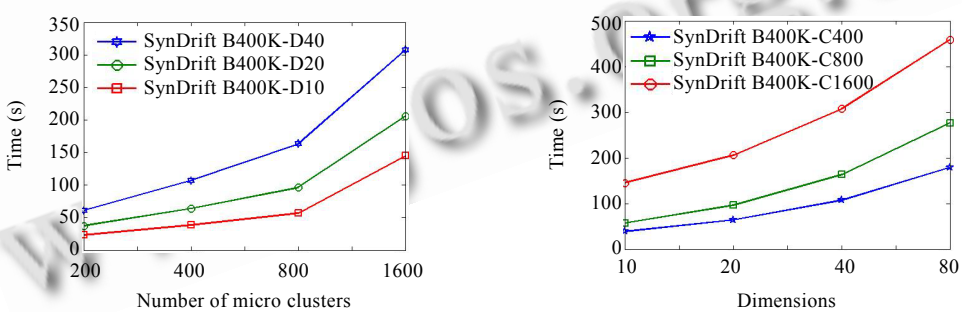
(a) Network intrusion detection dataset
(a) 网络入侵检测数据集

(b) Forest covertype dataset
(b) 森林覆盖类型数据集

Fig.4 Time comparison between EMicro and UMicro

图 4 EMicro 与 UMicro 的运行时间比较

图 5 显示了基于模拟数据的测试结果.模拟数据集可以获得任意的维度,并对初始化的微簇个数进行调整.首先,数据集的微簇个数从 200 变化到 1 600,并固定数据流的长度和维度,图 5(a)显示,EMicro 的处理时间随簇的个数呈近线性增长.其次,改变数据集的维度,从 10 变化到 80,并固定数据流的长度和簇个数,图 5(b)显示,EMicro 的处理时间随维度线性增长.因此,算法的可扩展性较强.



(a) Time vs. number of micro clusters
(a) 处理时间与微簇个数的对比

(b) Time vs. dimensions
(b) 处理时间与数据集维度的对比

Fig.5 The scalability of EMicro

图 5 EMicro 算法的可扩展性

4.4 离群点处理机制与参数影响

最后测试离群点处理机制的影响.我们使用 Syndrift(ε)$\varepsilon\in$[0.05,0.3]数据集,ε越大,说明数据流进化的速度越快,离群点出现的机会越多.EMicro-Merge 与 EMicro-Buffer 分别代表使用异常处理机制前后的聚类质量,实验

© 中国科学院软件研究所 http://www.jos.org.cn

结果如图 6 所示.可以看出,使用离群点处理机制能显著改善聚类结果质量.

影响 EMicro 聚类质量的两个主要参数是缓冲比α和衰减速率比λ_{ratio},通过实验比较,可以为两者的设置提供一个合理的参考值.我们在数据集 Syndrift 中设置了不同的α和λ_{ratio}来考察它们与平均概率簇质量 AQ 和 SSQ 的关系.我们先固定住λ_{ratio}=1,分别设置α为 2%,3%,5%,10%和 20%.见表 2,开始阶段,SSQ 和平均概率簇质量 AQ 随α增大而增大,但 AQ 增长最多的是在α从 3%变化到 5%.这说明当缓冲比α变大时,离群点在缓冲微簇中可以有更长的考察时间,致使提升为核心微簇的可能性变大,从而使得平均概率簇质量 AQ 不断增大.但平均概率簇质量增大是以增大 SSQ 为代价的,所以不同应用中可根据需要来设置α取得 SSQ 和 AQ 的折衷.当达到最大值后,AQ 的值有所下降.这是因为将过多的微簇用于离群点的监测,加快了核心微簇的频繁变动,导致聚类的性能与质量有所下降.这也进一步说明在离群点比例不是很高的情况下,过度地分配资源给缓冲微簇是不必要的.

Table 2 Accuracy impact of buffer-ratios

表 2 缓冲比对准确性的影响

α=2%		α=3%		α=5%		α=10%		α=20%	
SSQ	*AQ*	*SSQ*	*AQ*	*SSQ*	*AQ*	*SSQ*	*AQ*	*SSQ*	*AQ*
13.928	307.72	18.712	352.95	23.204	466.11	28.241	310.5	50.421	311.72

最后固定α=3%,分别设置λ_{ratio}为 0.7,0.8,0.9,1,1.1,1.2 和 1.3.从图 7 中可以看出,当λ_{ratio}<1 时有利于相对稳定的 Syndrift(ε=0.05)数据集.但当λ_{ratio}>1 时,则更加有利于进化特性比较明显 Syndrift(ε=0.3)数据集.

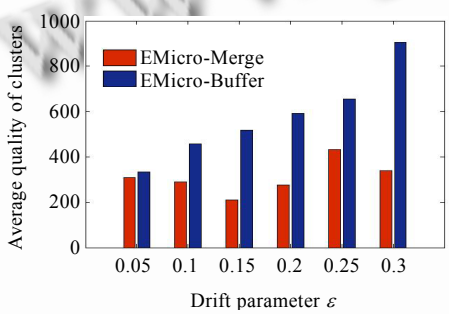


Fig.6 Accuracy impact of abnormal processing mechanism

图 6 异常处理机制对准确性的影响

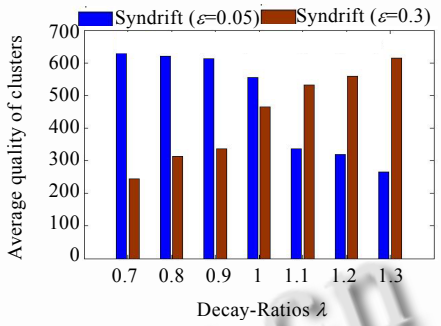


Fig.7 Accuracy impacts of decay-ratio

图 7 衰减速率比对准确性的影响

5 结论与展望

本文提出一种针对不确定数据流的聚类算法 EMicro,各元组具有存在级不确定性.该算法同时考虑元组间的距离因素与不确定性因素,更全面地描述了聚类结果质量.EMicro 算法的另一亮点在于提供了对离群点处理的充分支持.该算法在内存中保留了两份缓冲区,分别对应正常的微簇与潜在离群点的微簇,并提供了两个缓冲区的管理策略.实验结果表明,与现有算法相比,本文提出的 EMicro 算法在聚类质量、处理速度等方面均有优势.未来拟扩展该项技术,以解决滑动窗口模型下的聚类问题.

References:

[1] Babcock B, Babu S, Datar M, Motwani R, Widom J. Models and issues data stream systems. In: Proc. of the 21st ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems. Madison: ACM, 2002. 1−16.

[2] Aggarwal CC, Han JW, Yu PS. A framework for clustering evolving data streams. In: Proc. of the 29th Int'l Conf. on Very Large Data Bases. Berlin: Morgan Kaufmann Publishers, 2003. 81−92.

[3] Aggarwal CC, Yu PS. A framework for clustering uncertain data streams. In: Proc. of the 24th Int'l Conf. on Data Engineering. Cancún: IEEE, 2008. 150−159.

[4] Callaghan LO, Mishra N, Meyerson A, Guha S, Motwani R. Streaming-Data algorithms for high-quality clustering. In: Proc. of the 18th Int'l Conf. on Data Engineering. San Jose: IEEE, 2002. 685−694.

© 中国科学院软件研究所 http://www.jos.org.cn

[5] Zhu WH, Yin J, Xie YH. Arbitrary shape cluster algorithm for clustering data stream. Journal of Software, 2006,17(3):379−387 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/379.htm [doi: 10.1360/jos170379]

[6] Datar M, Gionis A, Indyk P, Motwani R. Maintaining stream statistics over sliding windows. In: Proc. of the 13th Annual ACM-SIAM Symp. on Discrete Algorithms. San Francisco: ACM, 2002. 635−644.

[7] Babcock B, Datar M, Motwani R, Callaghan LO. Maintaining variance and *k*-medians over data stream windows. In: Proc. of the 22nd ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems. San Diego: ACM, 2003. 234−243.

[8] Cao F, Estery M, Qian WN, Zhou AY. Density-Based clustering over an evolving data stream with noise. In: Proc. of the 6th SIAM Int'l Conf. on Data Mining. Bethesda: SIAM, 2006. 326−337.

[9] Kriegel HP, Pfeifle M. Density-Based clustering of uncertain data. In: Proc. of the 11th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Chicago: ACM, 2005. 672−677.

[10] Kriegel HP, Pfeifle M. Hierarchical density-based clustering of uncertain data. In: Proc. of the 5th IEEE Int'l Conf. on Data Mining. Houston: IEEE Computer Society, 2005. 689−692.

[11] Ngai WK, Kao B, Chui CK, Cheng R, Chau M, Yip KY. Efficient clustering of uncertain data. In: Proc. of the 6th IEEE Int'l Conf. on Data Mining. Hong Kong: IEEE Computer Society, 2006. 436−445.

[12] Zhang T, Ramakrishnan R, Livny M. Birch: An efficient data clustering method for very large databases. In: Proc. of the 1996 ACM SIGMOD Int'l Conf. on Management of Data. Montreal: ACM, 1996. 103−114.

[13] Aggarwal CC. On high dimension projected clustering of uncertain data streams. In: Proc. of the 25th Int'l Conf. on Data Engineering. Shanghai: IEEE, 2009. 1152−1154.

[14] Zhang C, Gao M, Zhou AY. Tracking high quality clusters over uncertain data streams. In: Proc. of the 1st Workshop on Management and Mining of Uncertain Data (MOUND 2009). Joint with ICDE 2009. Shanghai: IEEE, 2009. 1641−1648.

[15] Cormode G, McGregor A. Approximation algorithms for clustering uncertain data. In: Proc. of the 22nd ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems. Vancouver: ACM, 2008. 191−200.

[16] Kanagal B, Deshpande A. Online filtering, smoothing and probabilistic modeling of streaming data. In: Proc. of the 24th Int'l Conf. on Data Engineering. Cancún: IEEE, 2008. 1160−1169.

[17] Ré C, Letchner J, Balazinska M, Suciu D. Event queries on correlated probabilistic streams. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. Vancouver: ACM, 2008. 715−728.

[18] Newman DJ, Hettich S, Blake CL, Merz CJ. UCI repository of machine learning databases. http://archive.ics.uci.edu/ml/

[19] Aggarwal CC, Han JW, Wang JY, Yu PS. A framework for projected clustering of high dimensional data streams. In: Proc. of the 13th Int'l Conf. on Very Large Data Bases. Toronto: Morgan Kaufmann Publishers, 2004. 852−863.

附中文参考文献:

[5] 朱蔚恒,印鉴,谢益煌.基于数据流的任意形状聚类算法.软件学报,2006,17(3):379−387. http://www.jos.org.cn/1000-9825/17/379.htm [doi: 10.1360/jos170379]

张晨(1980－),男,上海人,博士生,主要研究领域为数据挖掘,数据流挖掘,不确定数据管理.

金澈清(1977－),男,博士,副教授,主要研究领域为数据流管理,不确定数据管理,移动数据管理.

周傲英(1965－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据库,数据挖掘,XML 数据管理,Web 搜索,P2P 计算和系统.

© 中国科学院软件研究所 http://www.jos.org.cn