

基于混合差分进化的滑动窗口 数据流聚类算法研究*

任永功, 胡志冬, 杨 雪

(辽宁师范大学 计算机与信息技术学院, 辽宁 大连 116029)

摘要: 针对传统的基于滑动窗口的数据流聚类算法存在的算法执行效率低、聚类质量较差等缺点, 提出了一种基于混合差分进化的滑动窗口数据流聚类算法。该算法将数据流聚类过程分为两个部分: 在线的时序窗口数据信息微簇特征向量生成和离线阶段的聚类优化。对在线生成的微簇进行微簇集合的更新与维护, 利用改进的粒子群算法对离线的微簇数据信息进行适应度值的计算, 将种群分为优势子种群和普通子种群, 然后利用个体适应度值和平均适应度值的判别来生成当前个体环境的最优候选解, 并迭代地对个体进行进化, 输出具有最优适应度值的聚类集合, 完成对数据流的聚类。仿真实验结果表明, 算法在对数据流执行聚类时具有较高的执行效率, 并且最后聚类的质量较好, 算法实用性强。

关键词: 混合差分进化; 滑动窗口; 数据流; 聚类

中图分类号: TP301.6

文献标志码: A

文章编号: 1001-3695(2014)04-1009-04

doi:10.3969/j.issn.1001-3695.2014.04.012

Research on sliding window data stream clustering algorithm based on hybrid differential evolution

REN Yong-gong, HU Zhi-dong, YANG Xue

(School of Computer & Information Technology, Liaoning Normal University, Dalian Liaoning 116029, China)

Abstract: In order to improve the execution efficiency and clustering quality of data stream clustering algorithm based on the sliding window, this paper presented a new a sliding window data stream clustering algorithm based on hybrid differential evolution. First, it divided data stream clustering process into two parts that were micro-clusters eigenvector of online timing-window and the offline clustering optimization, thus to update and maintenance the collection of micro-clusters which were generated online. Second, it calculated fitness value of offline micro-cluster data by using the improved particle swarm optimization and divided the population into the advantaged sub-population and the normal one. Then generated individual environment optimal candidate solutions by using individual and average fitness value of the discriminant. Finally, it performed iteration of individual evolution and output the optimal fitness value clustering collection.

Key words: hybrid differential evolution; sliding window; data flow; clustering

0 引言

数据聚类分析是数据挖掘领域中的一个重要研究课题, 其研究有利于对数据信息进行层次和模型分析, 尤其在数据流环境下, 数据元的增长呈指数等规模化变化^[1], 使得传统的聚类算法无法满足数据流处理的实时要求。因此基于数据流处理的聚类算法被广泛研究, 经典的算法如 CluStream^[2], 该算法将数据流的处理过程看做数据流的时序读取过程, 然后在数据处理周期内对数据进行聚类。由于其聚类半径的增长, 数据聚类的聚类精度较低, 且聚类的内存使用情况消耗较大, 因此文献[3]提出了基于连续属性的数据流聚类算法, 该算法的优势在于建立了数据流聚类的在线、离线框架, 并以数据流入点和流出点为数据流处理需求来调整滑动窗口, 但是由于数据快照窗口的失效数据未实时更新, 导致计算机负载较大。文献[4]提

出基于粒子群的聚类优化算法, 在时间滑动窗口模型中对数据流结构和粒子群质量进行优化迭代, 但是由于粒子群算法自身的缺陷, 易陷入早熟, 致使算法的执行解难以找到全局最优值。文献[5]提出了一种基于滑动窗口的数据流聚类方法, 该方法在占用窗口大小的次线性内存空间前提下, 及时保存最近数据记录的分布状况, 从而实现对滑动窗口内的数据进行聚类。为了保留数据流在聚类过程中的重要形态, 文献[6]提出了一种基于形态特征的数据流聚类方法。文献[7]在构造型神经网络的基础上引入一种新的数据流聚类相似性函数, 并根据滑动窗口模型数据流聚类的特点, 提出了基于构造型神经网络的滑动窗口模型数据流聚类算法。文献[8]提出了一种基于衰减滑动窗口的数据流聚类算法, 该算法把滑动窗口分为多个子窗口, 对每个子窗口上的数据进行聚类, 对聚类结果(各个候选微簇)集取代特定的记录进行处理。

收稿日期: 2013-04-03; **修回日期:** 2013-05-29 **基金项目:** 国家自然科学基金资助项目(60603047); 国家教育部留学回国人员科研启动基金资助项目辽宁省计划项目(2012232001); 辽宁省自然科学基金资助项目(201202119)

作者简介: 任永功(1972-), 男, 教授, 研究生导师, 博士, 主要研究方向为数据挖掘、图像处理技术等(415619436@qq.com); 胡志冬(1984-), 男, 硕士研究生, 主要研究方向为数据挖掘。

基于此,本文提出了一种基于混合差分进化的滑动窗口数据流聚类算法,通过在线阶段的微簇生成和离线阶段的混合差分进化算法的聚类优化,提高数据流聚类算法的执行效率。

1 基础定义

定义 1 窗口快照。设 t 时刻的数据窗口跨度为 p ,则在时段 $[t-p, p]$ 内最新到达的数据流 DB_i 为时间窗口 B 的一个快照,记为 $b_i(\text{size}) = |b_i|, b_i(\text{span}) = p$ 。

定义 2 时序滑动窗口。若快照窗口按照数据流的先后到达顺序构成时序数据流,记为 SB ,则令时序 i 时的时序滑块窗口为 $SB_i = \{b_{i-w+1}, b_{i-w+2}, \dots, b_i\}$ 。

若窗口基本数为 n ,则时间跨度 $sb_i(\text{span}) = n \times p$ 。

定义 3 时序衰减权重系数。设时刻 t 时的时序窗口衰减权重因子 ε 满足 $2^{-\lambda t} < \varepsilon, \lambda > 0$,则数据流的时序衰减权重系数 $W(t)$ 为

$$W(t) = v \times \sum_{i=0}^{t=t_c} 2^{-\lambda t} = \frac{v}{1-2^{-\lambda}} \quad (1)$$

其中: t_c 为当前的滑动窗口时间, v 为数据流的流速。

定义 4 数据流微簇。设当前数据流在时序滑动窗口中的微簇为 CF ,记为 $CF = \langle F, Q, n, RT_1, RT_2, RW \rangle$,其中数据集为 $\{X_1, X_2, \dots, X_n\}$, (F, Q) 为样本数据属性的一阶和二阶矩阵,数据流微簇的样本总数为 n ,当前数据流到达时间和失效时间为 (RT_1, RT_2) ,当前滑动窗口的大小为 RW ,则

$$F_j = \sum_{k=1}^n X_{kj}, Q_j = \sum_{k=1}^n (X_{kj})^2 \quad (2)$$

定义 5 样本聚类权重系数。若当前时序滑动窗口的数据流为 SB ,对于每个待识别的样本 Y ,其隶属于类别 w_i 的近邻样本总数为 k ,设当前样本总数为 m ,则对第 j 个近邻样本在进行聚类时,其样本聚类权重系数 $l(j)$ 为

$$l(j) = \frac{d_i^{-\alpha}}{\sum_{j=1}^k d_j^{-\alpha}} \quad (3)$$

其中: α 为幂指数。

定义 6 聚类类别判别函数。设数据集的样本类别为 ω_i ,待识别数据为 Y ,其数据集中近邻的数据中属于类别 ω_i 的样本为 X_i ,近邻样本总数为 N ,训练样本总数为 N_j ,近邻样本中隶属于类别 ω_i 的样本数为 N_i ,待识别数据 Y 的第 j 个近邻样本隶属于类别 ω_i 的类别判别函数 $G(Y_j, \omega_i)$ 为

$$G(Y_j, \omega_i) = \left[p_1 \times \sum_{j=1}^k l(x) \frac{\sum_{i=1}^r N(V_i)}{k} \right] + p_2 \times N + p_3 \times \frac{\sum_{i=1}^n (V_i)}{N_j} \quad (4)$$

其中: $\{p_1, p_2, p_3\}$ 为识别参数, $p_1 + p_2 + p_3 = 1$ 。

2 基于混合差分进化的滑动窗口数据流聚类算法

2.1 算法思想

设时序滑动窗口的训练数据集为 $LB = \{(\omega_s, x_i) | i = 1, 2, \dots, n; s = 1, 2, \dots, c\}$,其样本的类别数为 c ,样本的类别标示符为 ω_s ,当前的数据流为 DB 。若设当前时序滑动窗口中每个窗口快照达到的数据集为 DB_i ,待识别样本为 x_i ,其类别为 ω_i ,则有以下两个过程:

a) 时序滑动窗口的初始化。

设当前的快照窗口 B_i 中已达到数据集为 DB_{ij} ,记其数据

为 $A[i]$,则根据定义 2, $A[i]$ 包括 $(n+1)$ 个数据元组, n 为时间窗口数。

b) 时序滑动窗口的更新。

时序滑动窗口的更新主要涉及快照窗口数据信息的存储和失效数据信息的删除。若设 B_i 到达时 $B_{i-j} \in A[j], 1 \leq j \leq n+1, i-j > 0$,当 B_i 快照窗口内的数据流 DB_{ij} 被处理完时, $A[n+1]$ 元组被删除,并令 $A[j] = A[j+1]$,将 B_i 快照窗口的数据信息存储于 $A[j]$ 。

基于混合差分进化算法的滑动窗口数据流聚类算法主要包括在线聚类和离线聚类过程。由于数据流具有一定的流速,因此在线聚类主要是利用时序滑动窗口和快照窗口将数据流的粒度和流速信息进行微簇特征向量形式的存储,而离线阶段是对微簇特征向量的数据流粒度数据进行优化聚类。

2.2 在线聚类的微簇特征向量生成

微簇特征向量的生成主要是利用 DBSCAN 算法^[9]生成微簇的集合。具体的算法如下:

a) 设微簇表示为 num ,初始时令 $\text{num} = 0$ 。

利用 DBSCAN 算法进行微簇集合的生成。若设对象 p 满足簇半径 $r < p$ 时, $w \geq \beta\mu$,则令 p 为候选簇,记 $\text{num} + 1$,在对象集合中删除 p 。迭代地对当前时序滑动窗口的数据流对象进行初次分类。

b) 对时序滑动窗口 n 内的快照窗口 B_i ,若对象 p 满足 $r_i < r$,则该对象隶属于核心簇,其微簇的结构为 $(\overline{CF^2} + p^2, \overline{CF} + p, w + 1, t, \tau)$;否则属于孤立核心簇,更新簇集合。

c) 对于孤立核心簇,若 $w \geq \beta\mu$,则删除该簇,更新孤立核心簇数据集。

d) 若当前时序滑动窗口的数据流对象均被初次分簇,则聚类微簇的特征向量为 $(\overline{CF^2} \times 2^{-\lambda\Delta t}, \overline{CF} \times 2^{-\lambda\Delta t}, n \times 2^{-\lambda\Delta t}, t_i, \tau)$,其中 Δt 为微簇特征向量更新的时间间隔。

e) 设当前时刻 t_c 的快照窗口更新周期为 T ,令 $T \leq T_d < 2T$,若当前微簇 mc 满足 $|mc, t_i - t_c| \geq T_d$,则删除微簇 mc ;否则令权重系数 $W_c < \delta(t_c, t_a)$,且

$$\delta(t_c, t_a) = \frac{2^{-\lambda(t_c - t_a + T_d) - 1}}{2^{-\lambda T_d} - 1} \quad (5)$$

其中: t_c 为当前的时刻, t_a 为孤立核心簇的更新时间。

f) 将微簇数据集的特征向量和数据信息的粒度快照信息进行存储。

2.3 离线阶段的数据流聚类优化

本文在进行离线阶段的数据流聚类优化时,采取混合差分进化算法对微簇数据集的特征向量进行优化聚类。进化算法由于其具有良好的算法鲁棒性和简洁的可执行性,在优化问题求解中具有良好的表现。典型的算法有粒子群算法。

2.3.1 进化算法及其改进

粒子群算法主要是指粒子在空间维度中拥有特定的速度进行飞行,并且动态地对粒子的位置进行调整,以一定的个体认知和社会学习能力对当前环境中的最优解和群体最优解进行搜索。设粒子群的规模为 M ,空间维度为 D ,则第 i 个粒子在第 d 维空间中的位置集合为 $X_i = \{x_{i1}, x_{i2}, \dots, x_{iD}\}$,粒子的速度集合为 $V_i = \{v_{i1}, v_{i2}, \dots, v_{iD}\}$,个体的最优位置集合为 $P_i = \{p_{i1}, p_{i2}, \dots, p_{iD}\}$,种群的全局最优位置集合为 $P_g = \{p_{g1}, p_{g2}, \dots, p_{gD}\}$,则粒子 i 在第 $(t+1)$ 时刻的粒子速度和位置更新

策略为

$$\begin{cases} v_{id}^{(t+1)} = v_{id}^t + c_1 \times r_1 (p_{id}^t - x_{id}^t) + c_2 \times r_2 (p_{gd}^t - x_{id}^t) \\ x_{id}^{(t+1)} = x_{id}^t + v_{id}^{(t+1)} \end{cases} \quad (6)$$

其中: $\{c_1, c_2\}$ 为粒子的加速系数; $\{r_1, r_2\}$ 为 $[0, 1]$ 间的随机数; w 为惯性系数。

粒子群算法在进行全局寻优时,随着迭代计算量的增大,粒子的整体变化趋势趋缓,且趋势趋于一致,这将导致粒子的活动降低,从而导致粒子后期迭代计算时收敛较难。文献[10]提出了将惯性系数引入到粒子群算法中,对粒子的全局寻优能力进行调节,提高算法的执行效率。粒子的速度更新机制为

$$v_{id}^{(t+1)} = w \times v_{id}^t + c_1 \times r_1 (p_{id}^t - x_{id}^t) + c_2 \times r_2 (p_{gd}^t - x_{id}^t) \quad (7)$$

粒子群优化算法中主要是需要对当前求解的最优解是否为全局最优解进行判定,即对当前求解值的适应度函数值和上一时刻的适应度值进行比较,若其趋势是更优的,则需要对粒子的当前参数进行更新。更新时粒子在新环境中的适应度值决定了整个种群的进化趋势。基于此,文献[11]提出了新的权重更新策略,即设 $\Delta w = w_{\max} - w_{\min}$, 则

$$w(t) = w_{\max} - \frac{\Delta w \times t}{t_{\max}} \quad (8)$$

其中: $\{w_{\max}, w_{\min}\}$ 分别为最大设置权值和最小设置权值; t 为当前种群的迭代次数; $t_{\max}$ 为粒子的迭代总次数。

在进行粒子惯性函数赋值时,若其映射策略为线性递减时,则算法的极值点未必为真实的动态极值点,且较大的权值会影响当前的粒子以较大的更新量偏离当前环境的极值点,因此需要对粒子权值的映射策略进行改进。文献[12]提出了粒距的概念,设粒子在第 t 次迭代计算时第 j 个粒子在第 i 维空间中的粒距为 $d_{ij}(t)$, 则

$$d_{ij}(t) = |x_{ij}(t) - g_{\text{best}}(t)| \quad (9)$$

在实际的粒距计算时,可能由于粒子速度具有一定的聚类性,则粒子按照速度进行分类时会产生紧集合。当粒距较小时,粒子较易陷入局部极值中,造成收敛较难。因此在进行粒距分类时,记录粒子的聚类程度,当粒子聚类较高、粒距间距较小时,增大粒子的惯性系数,否则降低惯性系数。则有

$$\begin{cases} d_{\text{mean}}(t) = \frac{|\sum_{j=1}^n \sum_{i=1}^d d_{ij}(t)|}{n \times d} \\ d_{\text{max}}(t) = |\max[d_{ij}(t)]| \\ k = \frac{|d_{\text{max}}(t) - d_{\text{mean}}(t)|}{d_{\text{max}}(t)} \end{cases} \quad (10)$$

其中: $d_{\text{mean}}(t)$ 为粒子当前环境中的平均粒距; $d_{\text{max}}(t)$ 为最大粒距; k 为粒子当前环境中的聚类度,其值域为 $[0, 1]$, 则粒子权重系数映射策略为

$$\begin{cases} w = w(t) \times w_{\text{start}} & k \geq \alpha \\ w = w(t) \times \frac{1}{w_{\text{end}}} & k < \beta \end{cases} \quad (11)$$

其中: $\{\alpha, \beta\}$ 为惯性系数调整参数。一般取 $w_{\text{start}} = 0.95, w_{\text{end}} = 0.05$ 。

2.3.2 差分进化算法

目前的粒子群算法在进行大规模的数据聚类时,仅对内容算法在权系数上进行调整,在某种程度上已经受到本身算法框架的制约,无法避免适应度值最终趋于一致。虽然引入种群活性的提升策略,但是由于速度和位置更新机制的个体学习认知和社会学习能力制约,仍存在着早熟和易陷入局部极值点的缺

陷。基于此,提出了混合差分进化算法。差分进化算法是基于群体演化的优化算法,其基本框架源自于遗传算法,同时借助于单纯行算法的差分变异算子,具有一定的全局优化性能和数值稳定性。

差分进化算子包括变异、交叉和选择,常用 $DE/x/y/z$ 来标记不同的差分算法,其中 x 为基向量类型, y 为变异操作采用的差分向量个数, z 为交叉操作的类型。经典的差分算法为 $DE/\text{rand}/1/\text{bin}$, 在该算法中,第 i 个差分个体的变异操作需要从当前种群中随机选择三个不同的个体 $\{x_{r1}, x_{r2}, x_{r3}\}$, 且以随机的方式选出个体 x_{r1} 作为基向量, $\{x_{r2}, x_{r3}\}$ 的差向量按照一定的比例缩放后叠加在基向量上,即

$$z_{(i,d)} = x_{r1} + F \times (x_{r2} - x_{r3}) \quad (12)$$

其中: F 为缩放因子。

变异操作后,采用二元交叉操作对变异向量 z_i 和对应的原向量 x_i 进行操作,以生成最终的向量,即

$$u_{(i,d)} = \begin{cases} z_{(i,d)} & \text{rand} \leq C_R, d = r_{m_i} \\ x_{(i,d)} & \text{other} \end{cases} \quad (13)$$

其中: $\{x_{(i,d)}, z_{(i,d)}, u_{(i,d)}\}$ 分别为解向量 $\{x_i, z_i, u_i\}$ 的第 d 维分量; C_R 为交叉概率; r_{m_i} 是随机整数,用于确保最终向量不同于原向量。

2.3.3 混合差分进化算法

本文提出的混合差分进化算法以粒子群算法为母框架算法^[8],首先利用粒子群算法作用于整个种群,然后将所有个体按照平均适应度值分为两部分,将适应度值较差的子种群用粒子群算法进行进化,将适应度值高于平均适应度值的个体用差分进化算法进行进化,即

$$\begin{cases} v_{id}^{(t+1)} = w \times v_{id}^t + c_1 \times r_1 (p_{id}^t - x_{id}^t) + c_2 \times r_2 (p_{gd}^t - x_{id}^t) \\ x_{id}^{(t+1)} = x_{id}^t + v_{id}^{(t+1)} \end{cases} \quad (14)$$

$$z_{(i,d)}^{(k+1)} = x_{r1}^k + F \times (x_{r2}^k - x_{r3}^k) \quad (15)$$

$$u_{(i,d)}^{k+1} = \begin{cases} x_{id}^{(t+1)} & f_{\text{fitness}}^t < f_{\text{fitness}}^* \\ z_{(i,d)}^{(k+1)} & f_{\text{fitness}}^t \geq f_{\text{fitness}}^* \end{cases} \quad (16)$$

其中: 候选解 $u_i^{(k+1)} = \{u_{i1}^{(k+1)}, u_{i2}^{(k+1)}, \dots, u_{id}^{(k+1)}\}$ 和种群中的个体 $x_i^k = \{x_{i1}^k, x_{i2}^k, \dots, x_{id}^k\}$ 进行适应值对比,选取最优值保存为新的种群解。

2.3.4 混合差分进化的聚类优化算法

本文的离线阶段聚类优化主要利用混合差分进化算法,具体的算法步骤如下:

- 读取内存存储的微簇数据。
- 对于微簇 mc , 在簇半径内对微簇的特征向量随机地初始化其粒子位置和速度信息。
- 对微簇中待识别的数据对象计算其与粒子相对应的 k 个聚类中心的距离,计算当前粒子环境的权系数,并根据粒子的速度进行粒距分类。
- 根据粒距聚集度更新粒子的权重值。
- 对种群进行一次粒子群算法操作,利用以下策略计算各个体的适应度值:

$$f = \frac{\alpha d_1}{1 + \beta d_2} \quad (17)$$

$$d_1 = \frac{2 \sum_{i=0}^k \sum_{j=i+1}^k d(\omega_i, \omega_j)}{k(k-1)} \quad (18)$$

$$d_2 = \frac{1}{k} \times \sum_{i=0}^k \frac{\sum_{j=0}^{m_i} d(\omega_{ij}, \omega_i)}{m_i} \quad (19)$$

其中: $\{\omega_i, \omega_{ij}, m_i\}$ 分别为第 i 个聚类中心、第 j 个数据元组和数据总数; $\{\alpha, \beta\}$ 为类间距的调整权重。

f) 计算当前种群的平均适应度值, 根据适应度值将种群分为两个子种群。

g) 对于个体适应度大于平均适应度值的个体, 进行差分进化, 生成新的候选解; 对于个体适应度值小于平均适应度值的个体, 利用粒子群算法进行更新, 生成新的解。

h) 根据个体自适应度值的比较, 更新个体极值和全局极值, 并保存当前个体环境下的最新解。

i) 迭代地进行聚类优化, 输出最终的聚类集合。

3 实验仿真

实验环境为 Windows XP 系统, 2.66 GHz 奔腾四处理器, 512 MB 内存。算法运行平台为 VC++6.0 和 SQL Server 2005, 数据集来自 KDDCUP99 和 B400C20D40 的网络入侵数据流和在 MATLAB2012b 平台上人工生成的满足高斯分布的数据集组成, 共 494 020 条记录。实验初始的参数为 $\text{InitN} = 1000$, 数据流速为 $v = 1000$, 时间滑动窗口的衰减因子 $\lambda = 0.25$, 微簇半径设为 $R = 16$, 权重阈值设为 $\mu = 10$, 误差系数设为 $\beta = 0.001$, 并令 $T = 5$, 微簇的更新周期 $T_d = 9$ s, 滑动窗口的平均长度 $n = 5$, 进化计算时种群迭代次数为 100, 聚类中心为 20, 惯性权值的阈值为 $[0.4, 0.9]$, 加速因子为 2, 速度取值为 $[0, 1]$ 。为了对本文算法的有效性进行验证, 选取文献[3]中的 TSWDCluStream 算法对数据集 KDDCUP99 和 B400C20D40 进行处理, 具体的对比如图 1 所示。

从图 1 可以看出, 本文算法对数据集 KDDCUP99 进行处理时, 其效率要优于 TSWDCluStream 算法。这主要是本文算法对聚类效果进行了进化优化, 且本文算法在两种数据集上进行测试时, 其执行时间比均在 95% ~ 100% 之间。

针对本文算法的聚类效果与数据流窗口长度的关系进行了对比仿真, 数据集仍采用网络入侵数据集 KDDCUP99 和 B400C20D40, 具体的对比如图 2 和 3 所示。从图 2 和 3 可以看出, 两种算法的执行时间随着数据流窗口数的增多均有增长, 但是本文算法相对于文献[3]的算法, 由于引入了混合差分进化算法, 对聚类中心进行了优化, 并且具有较强的算法适应性, 因此该算法在数据流增长时执行效果较优。

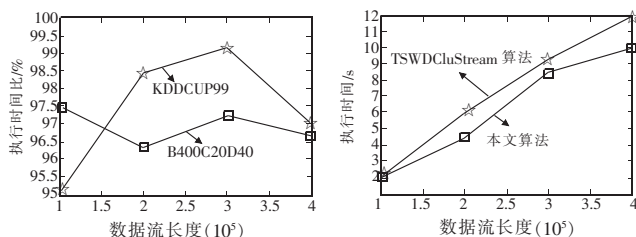


图1 两种算法的执行时间比

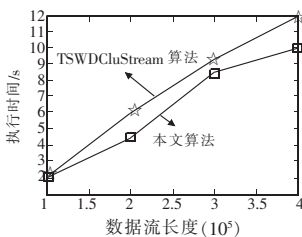


图2 两种算法在 KDDCUP99 数据集的执行效率对比

取快照窗口为 B_i , 在不同周期 T 下对算法的数据流处理时内存使用情况进行仿真, 具体如图 4 所示。

从图 4 可以看出, 内存使用在 343 KB 左右时, 算法对数据流进行处理的曲线区域平稳, 这表明本文算法在常规计算机硬

件环境下, 具有较好的适用性。

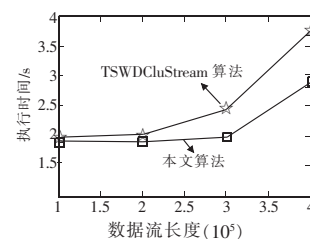


图3 两种算法在 B400C20D40 数据集的执行效率对比

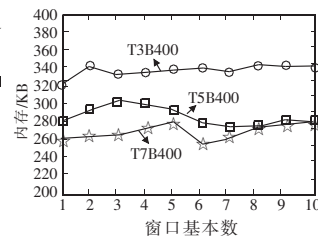


图4 快照窗口内不同周期的数据流处理内存对比

4 结束语

通过对数据流的快速聚类算法进行研究, 提出了在时序滑动窗中对快照窗口的数据信息利用混合差分进化算法在离线阶段进行聚类优化, 而在在线阶段仅对数据流的微簇特征向量和粒度信息进行微簇的生成、更新和在线存储。该算法有效地利用滑动窗口模型, 将在线环节与离线环节同时进行, 避免了数据流规模过大时影响算法的执行效率。同时该算法引入混合差分进化算法, 改善了粒子群算法的早熟现象, 在一定程度上增强了进化算法寻找最优解的能力, 并且对于粒子群算法进行了权重映射改进。由于在聚类优化时, 每次都进行个体适应度值和平均适应度值的比较, 进而形成优势子群和普通子群, 并采取不同的策略进行种群候选解的生成, 保证了每次迭代过程中解都具有更优的搜索能力, 提高了算法的性能。实验仿真表明, 该算法的执行效率高, 对内存的依赖较小, 适应性较强。

参考文献:

- [1] SUN Y T, LU Y S. An overview of stream data mining [J]. Computer Science, 2007, 34(1): 1-5.
- [2] AGGARWAL C C, HAN Jia-wei, WANG Jian-yong, et al. A framework for clustering evolving data streams [C] // Proc of the 29th International Conference on Very Large Data Bases. [S. l.]: VLDB Endowment, 2003: 81-92.
- [3] 张忠平, 王浩, 薛伟, 等. 动态滑动窗口的数据流聚类方法 [J]. 计算机工程与应用, 2011, 47(7): 135-138.
- [4] 肖裕权, 周肆清. 基于粒子群优化算法的数据流聚类算法 [J]. 计算机技术与发展, 2011, 21(10): 43-47.
- [5] 常建龙, 曹锋, 周傲英. 基于滑动窗口的进化数据流聚类 [J]. 软件学报, 2007, 18(4): 905-918.
- [6] 吴学雁, 黄道平. 基于形态特征的数据流聚类方法研究 [J]. 计算机工程, 2011, 37(13): 46-49.
- [7] 陈荣晖, 王伦文. 一种新的滑动窗口模型数据流聚类方法 [J]. 小型微型计算机系统, 2010, 31(12): 2355-2359.
- [8] 朱琳, 刘晓东, 朱参世. 基于衰减滑动窗口数据流聚类算法研究 [J]. 计算机工程与设计, 2012, 33(7): 2659-2662.
- [9] 陈燕, 耿国华, 郑建国. 一种改进的基于密度的聚类算法 [J]. 微机发展, 2005, 15(3): 17-19.
- [10] 黄泽霞, 俞攸红, 黄德才. 惯性权自适应调整的量子粒子群优化算法 [J]. 上海交通大学学报, 2012, 46(2): 228-232.
- [11] 粒距, 左旭坤, 苏守宝. 基于粒距和动态区间的粒子群权值调整策略 [J]. 计算机应用, 2010, 30(9): 2286-2289.
- [12] YALLG K, MAGINU K, NOMURA H. Cultural algorithm-based on quantum-behaved particle swarm optimization [J]. International Journal of Computer Mathematics, 2010, 87(10): 2143-2157.