基于编辑距离的序列聚类算法的优化

孙启航,杨鹤标
(江苏大学 计算机科学与通信工程学院,江苏 镇江 212013)

摘 要:现有的很多序列聚类算法都是基于“局部特征可以代表整个序列”的假设,在实际应用中不对序列局部相似性和全局相似性加以区分,这对于存在子模式的序列聚类是适用的,如基因序列和蛋白质序列。但是对于不存在子模式的序列,如对临床行为序列、用户购买行为序列进行聚类时,用基于全局相似性度量的聚类方法更为恰当。针对不存在子模式的序列聚类的需要,采用编辑距离作为序列相似性计算方法,在二分 K 均值算法的基础上,提出了利用编辑距离上下界以及通过前缀子序列进行剪枝的序列聚类算法 PSClu。该算法能有效过滤编辑距离的计算量。实验结果表明,PSClu 能有效减少编辑距离的直接计算,具有较好的聚类效率和聚类质量。

关键词:序列聚类;编辑距离;二分 K 均值;序列相似性
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2018)03-0109-05
doi:10.3969/j.issn.1673-629X.2018.03.023

Optimizing of Sequence Clustering Algorithm Based on Edit Distance

SUN Qi-hang, YANG He-biao
(School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang 212013, China)

Abstract: Many of the existing sequence clustering algorithms are based on the assumption that local features can represent the entire sequence. The local similarity and the global similarity are not distinguished in practical applications, which is suitable for sequence clustering with child patterns, such as gene sequences and protein sequences. But, when clustering the sequences without the subpatterns, such as the clinical behavior sequences, customer purchasing sequences, it is more appropriate to utilize the clustering algorithm based on global similarity measure. To deal with these problems, with the edit distance as sequence similarity calculation method, on the basis of the binary k-means algorithm, we propose the PSClu algorithm which can effectively filter the computation of edit distance. The experiments show that PSClu can effectively reduce the direct calculation of edit distance with good clustering efficiency and quality.

Key words: sequence clustering; edit distance; binary k-means; sequence similarity

1 概 述

序列聚类的基础问题就是序列的相似性度量[1]。直观看来,当序列中对应位置的对象的值相似时,才被认为是相似的。在现有的研究中,根据度量范围的不同,序列相似性度量算法可分为如下两类:

(1)局部序列相似性度量。在生物信息学研究中,DNA 序列或蛋白质序列中往往存在着最能体现序列整体特征的关键片段,这些关键片段在刻画序列特点时占有相当高的比重,因此可以用来度量序列的相似性[2]。基于局部序列的相似性度量算法,核心就是要提取出能表征不同序列的局部序列,从而在局部序列的基础上度量全序列的整体相似性。

(2)全序列相似性度量。序列的特征通过整个序列来体现,这时度量序列的相似性必须从序列的全局考虑。例如有 5 个患者 P_1、P_2、P_3、P_4、P_5在不同时间点的临床行为序列如图 1 所示。

Patient P_1's clinical sequence:	14	21	30	34	26	67	90	45	70	29
Patient P_2's clinical sequence:	33	21	62	92	17	76	19	43	70	29
Patient P_3's clinical sequence:	33	95	62	34	17	67	19	45	57	56
Patient P_4's clinical sequence:	72	72	54	54	46	68	53	70	57	56
Patient P_5's clinical sequence:	11	38	59	80	22	22	16	65	57	56

图 1 患者的临床行为序列

尽管患者 P_3与患者 P_4、P_5的临床行为序列有一致的频繁子模式 57、56,但是患者 P_3与 P_2、P_3与 P_1、P_1与 P_2在对应时刻的相同医疗行为数量分别为 4,3,3,因此 P_1、P_2、P_3整体上的临床行为更相似。编辑距离的相似性度量算法能够准确反映序列全局相似度[3],但其

收稿日期:2017-04-05 **修回日期**:2017-08-15 **网络出版时间**:2017-12-05
基金项目:国家自然科学青年基金项目(61502208)
作者简介:孙启航(1992-),男,硕士,研究方向为数据挖掘;杨鹤标,教授,硕导,研究方向为软件方法、数据挖掘。

时间复杂度相对较高。

2 相关技术

K 均值是基于划分的聚类方法[4-6],该算法首先指定 K 值,并在数据集中选择 K 个点作为簇的初始质心。然后将每条序列都分配到距离它最近的质心,当完成一遍划分后,更新每个簇的质心。重复分配序列的过程,直到质心不再变化,聚类结果才是稳定的,最终所有数据都被划分到 K 个簇中。

相比于 K 均值聚类,二分 K 均值[7]避免了两个序列之间距离的直接计算,时间复杂度是线性增长的。序列的二分 K 均值聚类算法可作如下描述[8]:把所有序列初始化为一个簇加入簇表;从簇表中选取一个总体相似度水平最低的簇 C;利用 K-means 方法将簇 C 二分为 C_1 和 C_2;将 C_1 和 C_2 加入簇表中;重复以上步骤,直到产生 K 个簇时停止。

编辑距离[9]是计算两个序列之间相似度的算法。求解的常用思路是动态规划法[10],即对于两个序列 $A=\{a_1,a_2,\cdots,a_m\}$ 和 $B=\{b_1,b_2,\cdots,b_n\}$,用 $\mathrm{dis}(i,j)$ 表示序列 A 的前 i $(0\leqslant i\leqslant m)$ 个项通过编辑转化为序列 B 的前 j $(0\leqslant j\leqslant n)$ 个项所需的最小代价。序列 A 和 B 的相似度计算方法具有子结构和子问题重叠性质,可以通过不断地递归求解 $\mathrm{dis}(i,j)$ 最优解得到。具体的相似度计算步骤如下:

(1)参数初始化。

$\mathrm{distance}(A,B)=\mathrm{dis}(m,n)$;$\mathrm{dis}(0,0)=0$;

$\mathrm{dis}(0,j)=j$;$\mathrm{dis}(i,0)=i$

(2)递归计算序列相似度。

$$\mathrm{dis}(i,j)=\min\begin{cases}\mathrm{dis}(i-1,j)+1\\ \mathrm{dis}(i,j-1)+1\\ \mathrm{dis}(i-1,j-1)+k(i,j)\end{cases},\quad 1\leqslant i\leqslant m,1\leqslant j\leqslant n \tag{1}$$

其中,当 $a_i=b_i$ 时,$k(i,j)=0$,否则 $k(i,j)=1$。

通过在动态规划矩阵中对式(1)的计算,矩阵右下角的 $\mathrm{dis}(m,n)$ 可以通过不断得到子问题的最优解来迭代解决,最终解得两个序列的编辑距离。

3 带剪枝策略的序列聚类算法

3.1 基于全序列比对的编辑距离上下界

定义 1:序列是由项表中的项目组成的有序项目集合,记为 $S=\{a_1,a_2,\cdots,a_n\}$,其中 $a_k\in E(1\leqslant k\leqslant n)$,$a_k$ 称为序列中的项。T 为临床行为项目表中项目的个数,$T=|E|$。序列 S 的长度为 n,$n=|S|$。

定义 2[11]:对于序列 $S=\{a_1,a_2,\cdots,a_n\}$,n_i 表示项在序列 S 中出现的次数$(1\leqslant i\leqslant T)$,$T(S)=\{n_1,n_2,\cdots,n_T\}$ 称为序列 S 的频数列表,称之为标签。对于两个序列 S_1 和 S_2,分别对应 $\{n_1^{S_1},n_2^{S_1},\cdots,n_T^{S_1}\}$ 和 $\{n_1^{S_2},n_2^{S_2},\cdots,n_T^{S_2}\}$ 两个标签,S_1 和 S_2 的标签距离表示为:

$$\mathrm{TD}(S_1,S_2)=\max\Big\{\sum_{i=1}^{T}I_i^P(n_i^{S_1}-n_i^{S_2}),\ \sum_{j=1}^{T}I_j^N(n_j^{S_2}-n_j^{S_1})\Big\} \tag{2}$$

其中,I_i^P 和 I_j^N 均为示性函数。当 $n_i^{S_1}>n_i^{S_2}$ 时,$I_i^P=1$,否则 $I_i^P=0$;当 $n_j^{S_2}>n_j^{S_1}$ 时,$I_j^N=1$,否则 $I_j^N=0$。

标签距离和编辑距离的关系如定理 1 所示,这给后文聚类算法的剪枝策略提供了理论基础。

定理 1:对于给定的两个临床行为序列 S_1 和 S_2,标签距离 $\mathrm{TD}(S_1,S_2)$ 与两条序列的长度之和 $|S_1|+|S_2|$ 分别是编辑距离 $\mathrm{ED}(S_1,S_2)$ 的下限和上限,即 $\mathrm{TD}(S_1,S_2)\leqslant\mathrm{ED}(S_1,S_2)\leqslant|S_1|+|S_2|$。

$\mathrm{TD}(S_1,S_2)\leqslant\mathrm{ED}(S_1,S_2)$ 的证明见文献[11]。下面证明 $\mathrm{ED}(S_1,S_2)\leqslant|S_1|+|S_2|$。

在 S_1 和 S_2 的求解矩阵中,通过动态规划将 S_1 编辑为 S_2 时,从矩阵最右下角的 (m,n) 位置开始回溯,直到$(0,0)$时终止。每一次回溯或者是纵坐标减 1,或者是横坐标减 1,或者是横坐标和纵坐标均减 1,最多经过 $m+n$ 步可以走到$(0,0)$坐标,所以 $\mathrm{ED}(S_1,S_2)\leqslant|S_1|+|S_2|$。故可推出定理 1,$\mathrm{TD}(S_1,S_2)\leqslant\mathrm{ED}(S_1,S_2)\leqslant|S_1|+|S_2|$。

由定理 1 可以推出下列结论:

推论 1:给定三个临床行为序列 S_1、S_2、S_3,若 $\mathrm{TD}(S_1,S_2)\geqslant|S_1|+|S_3|$,则有 $\mathrm{ED}(S_1,S_2)\geqslant\mathrm{ED}(S_1,S_3)$。由定理 1 得:$\mathrm{ED}(S_1,S_2)\geqslant\mathrm{TD}(S_1,S_2)\geqslant|S_1|+|S_3|\geqslant\mathrm{ED}(S_1,S_3)$,故可得推论 1。

定理 2:给定三条临床序列 S_1、S_2、S_3,S_1 与 S_2 的编辑距离 $\mathrm{ED}(S_1,S_2)$ 已知,若 $\mathrm{ED}(S_1,S_2)\geqslant2\times(|S_1|+|S_3|)$,则 $\mathrm{ED}(S_1,S_3)\leqslant\mathrm{ED}(S_2,S_3)$。

证明:由于 $|S_1|+|S_3|\geqslant\mathrm{ED}(S_1,S_3)$,所以 $\mathrm{ED}(S_1,S_2)\geqslant2(|S_1|+|S_3|)\geqslant2\mathrm{ED}(S_1,S_3)$,通过移项得到 $\mathrm{ED}(S_1,S_2)-\mathrm{ED}(S_1,S_3)\geqslant\mathrm{ED}(S_1,S_3)$。由编辑距离的定义可知其满足三角不等式两边之差小于第三边[12],因而有 $\mathrm{ED}(S_2,S_3)\geqslant\mathrm{ED}(S_1,S_2)-\mathrm{ED}(S_1,S_3)$,所以 $\mathrm{ED}(S_2,S_3)\geqslant\mathrm{ED}(S_1,S_3)$。

计算序列编辑距离的时间复杂度为 $O(m\times n)$[13],计算标签距离的时间复杂度同为 $O(m\times n)$。对于待划分的序列和两个簇的质心,如果符合推论 1 和定理 2 中的条件,则可以仅通过计算标签距离来划分数据点,从而有效降低序列之间编辑距离的计算量。

3.2 基于前缀子序列的相似度比较

若推论 1 和定理 2 中的两个前提条件都不能满足,也有可能不需要进行两条序列的全序列比较,转而通过计算序列的部分编辑距离来衡量它们的临近度。对于给定的两个序列 $A=\{a_1,a_2,\cdots,a_m\}$ 和 $B=\{b_1,$

$b_2,\cdots,b_n\}$,假设其动态规划矩阵为 $\boldsymbol{T}$($\boldsymbol{T}$ 为一个 $m+1$ 行、$n+1$ 列的矩阵),则可以推导出下面的性质:

性质 1:$|\ \mathrm{dis}(i,j)-\mathrm{dis}(i-\Delta_i,j-\Delta_j)\ |\leqslant 1$。

其中,$\mathrm{dis}(i,j)$ 表示动态规划矩阵中第 i 行第 j 列的值,$0\leqslant i\leqslant m$,$0\leqslant j\leqslant n$,$\Delta_i,\Delta_j\in(0,1)$,并且 $i-\Delta_i\geqslant 0$,$j-\Delta_j\geqslant 0$。

证明:计算两条序列编辑距离的动态规划矩阵是由 $m+n-1$ 个初始状态 $\mathrm{dis}(i,j)=0$、$\mathrm{dis}(0,j)=j$,$\mathrm{dis}(i,0)=i(0\leqslant i\leqslant m,0\leqslant j\leqslant n)$ 和式(1)生成的,若 $a_i=b_j$,则 $k(i,j)=0$,否则 $k(i,j)=1$。由归纳法证明 $\mathrm{dis}(i,j)-\mathrm{dis}(i-\Delta_i,j-\Delta_j)$ 不会超过 1,性质 1 得证。

由性质 1 表明,在反映编辑距离的动态规划矩阵中,任何一个坐标对应的值减去紧邻其左边、上边或者左上角的值,只会有 0 和 1 两种结果,不会超过 1。图 2 所示的动态规划矩阵揭示了性质 1,且由性质 1 可以证明定理 3。

dis(*i*, *j*)	$	a	n	i	m	a	l	s
$	0	1	2	3	4	5	6	7
b	1	1	2	3	4	5	6	7
i	2	2	2	2	3	4	5	6
c	3	3	3	3	3	4	5	6
y	4	4	4	4	4	4	5	6
c	5	5	5	5	5	5	5	6
l	6	6	6	6	6	6	5	6
e	7	7	7	7	7	7	6	6

图 2　序列 animals 与 bicycle 的动态规划矩阵

定理 3:对于给定的两条序列 $A=\{a_1,a_2,\cdots,a_m\}$ 和 $B=\{b_1,b_2,\cdots,b_n\}$ $(m>n)$,其最小编辑代价的动态规划矩阵为 $\boldsymbol{T}$,则当 $i\leqslant j\leqslant n$ 时,存在 $\mathrm{dis}(i,i)\leqslant\mathrm{dis}(j,j)$。

证明定理 3 只要证明 $\mathrm{dis}(i,i)\leqslant\mathrm{dis}(i+1,i+1)$ 即可。根据动态规划矩阵 $\boldsymbol{T}$ 的生成过程,可以对以下三种情况进行分析:

(1)若 $\mathrm{dis}(i+1,i+1)=\mathrm{dis}(i,i)+k(i+1,i+1)$,因为 $k(i+1,i+1)$ 的取值为 0 或 1,所以 $\mathrm{dis}(i+1,i+1)\geqslant\mathrm{dis}(i,i)$。

(2)若 $\mathrm{dis}(i+1,i+1)=\mathrm{dis}(i+1,i)+1$,根据性质 1 可得 $|\ \mathrm{dis}(i+1,i)-\mathrm{dis}(i,i)\ |\leqslant 1$。分三种情况讨论:

① $\mathrm{dis}(i+1,i)-\mathrm{dis}(i,i)=1$,此时 $\mathrm{dis}(i+1,i+1)=\mathrm{dis}(i,i)+2$,$\mathrm{dis}(i,i)<\mathrm{dis}(i+1,i+1)$;

② $\mathrm{dis}(i+1,i)-\mathrm{dis}(i,i)=0$,此时 $\mathrm{dis}(i+1,i+1)=\mathrm{dis}(i,i)+1$,$\mathrm{dis}(i,i)<\mathrm{dis}(i+1,i+1)$;

③ $\mathrm{dis}(i+1,i)-\mathrm{dis}(i,i)=-1$,此时 $\mathrm{dis}(i+1,i+1)=\mathrm{dis}(i,i)$。

(3)若 $\mathrm{dis}(i+1,i+1)=\mathrm{dis}(i+1,i)+1$,参考情况(2)中的证明过程,仍可证得 $\mathrm{dis}(i+1,i+1)\geqslant\mathrm{dis}(i,i)$。

综上所述,有 $\mathrm{dis}(i,i)\leqslant\mathrm{dis}(i+1,i+1)$。故定理 3 得证。

定理 3 表明,在对两个序列进行全序列编辑距离的计算时,其长度相等的前缀子序列具有随着子序列中项的增加而编辑距离非递减的特性。例如图 2 中两种不同灰度的阴影部分所显示的,有 $\mathrm{ED}(an,bi)\leqslant\mathrm{ED}(ani,bic)$。

在临床行为异常检测中,需要将待检序列 S 和两个簇的质心 C_1、C_2 进行编辑距离的比较,若已知 S 和 C_1 的编辑距离 $\mathrm{ED}(S,C_1)$,又知道 S 和 C_2 某个等长前缀子序列的编辑距离比 $\mathrm{ED}(S,C_1)$ 大,那么不用计算 S 和 C_2 的全序列编辑距离就可以将 S 划入 C_1 所代表的簇中。对于长度不同的临床序列,通过定理 4,当在聚类过程中的编辑距离比较时,其中一个编辑距离只需求出等长前缀子序列的部分即可。

定理 4:设有三条序列 S、P、Q,其中 $S=\{s_1,s_2,\cdots,s_m\}$,$P=\{p_1,p_2,\cdots,p_n\}$ $(m>n)$,S 作为待分配序列需要比较它与 P、Q 的编辑距离的大小。若 S、Q 的编辑距离 $\mathrm{ED}(S,Q)$ 已知,且存在一个 $k\leqslant n$ 使得不等式 $\mathrm{ED}(s_1,s_2,\cdots,s_k,p_1,p_2,\cdots,p_k)-(m-n)\geqslant\mathrm{ED}(S,Q)$ 成立,则不等式 $\mathrm{ED}(S,P)\geqslant\mathrm{ED}(S,Q)$ 成立。

证明:设 S 和 P 的最优动态规划矩阵为 $\boldsymbol{T}$,将性质 1 中的绝对值符号去掉可以得到 $-1\leqslant\mathrm{dis}(i,j)-\mathrm{dis}(i-\Delta_i,j-\Delta_j)\leqslant 1$,故可以推得 $\mathrm{dis}(m,n)+1\geqslant\mathrm{dis}(m-1,n)$,$\mathrm{dis}(m-1,n)+1\geqslant\mathrm{dis}(m-2,n)$,$\cdots$,$\mathrm{dis}(n+1,n)+1\geqslant\mathrm{dis}(n,n)$,将这 $m-n$ 个不等式合并可得 $\mathrm{dis}(m,n)\geqslant\mathrm{dis}(n,n)-(m-n)$。根据定理 3,若存在一个 $k\leqslant n$,则有 $\mathrm{dis}(k,k)\leqslant\mathrm{dis}(n,n)$。因此,若 $k\leqslant n$ 且 $\mathrm{ED}(s_1,s_2,\cdots,s_k,p_1,p_2,\cdots,p_k)-(m-n)\geqslant\mathrm{ED}(S,Q)$ 成立,可得 $\mathrm{ED}(S,P)=\mathrm{dis}(m,n)\geqslant\mathrm{dis}(n,n)-(m-n)\geqslant\mathrm{dis}(k,k)-(m-n)\geqslant\mathrm{ED}(S,Q)$。

定理 4 的推导表明,在比较待检序列到簇的距离时,有时并不需要计算 S 和 Q 的全序列编辑距离就可以与 $\mathrm{ED}(S,Q)$ 进行比较,这大大降低了二分 K 均值算法在计算临床序列在计算编辑距离上的开销。

3.3　序列聚类算法的实现过程

对于数据集中的 t 条序列,其平均长度为 L,在以编辑距离为临近度度量的前提下,如果采用两两比较的方式来计算序列的编辑距离,算法的时间复杂度为 $O(t^2L^2)$,这严重影响了算法的聚类效率。为降低算法的复杂度,减少聚类时间,从以下几个方面来考虑:

(1)在聚类算法的选取上,选取二分 K 均值的方法。相比于 K 均值聚类,二分 K 均值避免了两个序列之间距离的直接计算,时间复杂度与数据集大小不是

呈指数增长,而是线性增长的。

(2)在处理簇间数据点的移动时,采取计算序列和簇的标签距离的方式避免不必要的计算。假设在应用二分 K 均值算法时,当前的两个质心是 C_1 和 C_2, S 表示序列数据集中的任意一条序列,且两个质心之间的编辑距离 ED(C_1,C_2)已知。在寻找离 S 点最近的簇时,需要计算 ED(S,C_1) 和 ED(S,C_2) 并比较这两者的大小。由推论 1 和定理 2 可知,当 Max{ TD(S,C_1),ED(C_1,C_2) /2} ≥ | C_2 | + | S |时, ED(S,C_2) ≤ ED(S,C_1) ,序列 S 放入 C_2 所代表的簇;当 Max{ TD(S,C_2) , ED(C_1,C_2) /2} ≥ | C_1 | + | S |时,有 ED(S,C_1) ≤ ED(S,C_2) ,序列 S 放入 C_1 所代表的簇。

若上述条件都不满足,才需要计算 ED(S,C_1) 和 ED(S,C_2) 的值。但是,也有可能只需计算 S 与其中一个质心的等长前缀子序列的编辑距离而不必计算全部就可以比较它们的大小。步骤(3)的剪枝策略描述了 S 到两个质心的编辑距离比较方法。

(3)对于临床行为序列数据集中的每一个序列 S ,创建一个维数为 $2T$ 的概貌向量 $\boldsymbol{V}_s = \{n_1, n_2, \cdots, n_T, d_1, d_2, \cdots, d_T\}$,其中 $n_i(1 \leq i \leq T)$ 是 S 的标签向量中的值,表示临床行为项表中第 i 个项在 S 中出现的次数, d_i 表示序列 S 中的项 e_i 到序列第一个项的距离之和。概貌向量在序列标签的基础上增加了反映序列中项的位置信息的维度,如果两条临床序列差异很大,它们所对应的概貌向量的曼哈顿距离也会很大[14]。

分析定理 4 可以发现,若想通过该定理减少编辑距离的计算复杂度,需要事先知道待检序列与哪一个簇的质心的编辑距离较小,通过计算三个序列概貌向量的曼哈顿距离来帮助判断。序列概貌向量的曼哈顿距离在很大概率上能帮助确定编辑距离的大小,从而有效决定待检临床序列与簇的质心编辑距离的计算顺序,最终达到简化计算序列临近度时间开销的目的。

3.4 带剪枝策略的二分 K 均值序列聚类算法

综合上一节的三点陈述,带剪枝策略的二分 K 均值临床序列聚类算法 PSClu(clustering algorithm based on similarity of prefix sequence)描述如下:

算法 1:带剪枝策略的临床序列聚类算法 PSClu。

输入:含有 t 条临床行为数据序列的数据集 SequenceSet = {S_1,S_2,…,S_t},参数 k

输出:序列的 k 个集合

/*起始状态下,将原始序列数据集 SequenceSet 初始化为一个簇并放入簇表 ClusterList 中,此时簇的个数 ClusterCount = 1 */

ClusterList←SequenceSet;ClusterCount = 1;

for(i = 1;i≤t;i++)

将序列数据集 SequenceSet 扫描一趟,生成 S_i 的标签 T(S_i) 以及概貌向量 V_{S_i};

end for;

while(ClusterCount<k)

从簇表 ClusterList 中选一个内部相似程度最差的簇 C;

分别将簇 C 中的最长序列和最短序列作为两个簇的质心 CO_1 和 CO_2;

计算两个质心的编辑距离 ED(CO_1, CO_2) 以及概貌向量 VCO_1 和 VCO_2;

while(簇 C 中所有序列不会在 CO_1 和 CO_2 两个质心代表的簇之间发生移动)

for(属于簇 C 的每一条序列 S')

if(Max{ TD(S', CO_1) , ED(CO_1, CO_2) /2} ≥ | CO_2 | + | S' |)

将序列 S' 放到以 CO_2 为质心的簇 C_2 中;

else if(Max{ TD(S', CO_2) , ED(CO_1, CO_2) /2} ≥ | CO_1 | + | S' |)

将序列 S' 放到以 CO_1 为质心的簇 C_1 中;

else if(L_1 ($V_{S'}$, V_{CO_1}) ≤ L_1 ($V_{S'}$, V_{CO_2}))

计算 ED(S', CO_1) ;

if(S' 和 CO_2 的等长前缀子序列满足定理 4 前提条件中的不等式)

终止 S' 与 CO_2 编辑距离的计算,并将序列 S' 放到以 CO_1 为质心的簇 C_1 中;

else

继续利用动态规划矩阵算完 ED(S', CO_2) ,然后确定将 S' 放到哪个簇;

else

计算 ED(S', CO_2) ;

if(S' 和 CO_1 的等长前缀子序列满足定理 4 前提条件中的不等式)

终止 S' 与 CO_1 编辑距离的计算,并将序列 S' 放到以 CO_2 为质心的簇 C_2 中;

else

继续利用动态规划矩阵算完 ED(S', CO_1) ,然后确定将 S' 放到哪个簇;

end for;

更新两个簇的质心 CO_1 和 CO_2;

end while;

ClusterCount = ClusterCount+1;

将 C_1 和 C_2 添加到簇表 ClusterList 中;

end while;

4 实验结果分析与验证

为了验证 PSClu 算法在对序列进行聚类分析时的性能,使用 Java 语言编码,并在人工合成的序列数据集上与其他算法进行对比实验。所有实验均在主频为 2.8 GHz,操作系统为 Windows7,可用内存为 3.24 G 的 PC 机上实现。

4.1 实验数据集的生成

类似文献[15]中实验所使用的方法,该课题合成数据集的方法如下:给定项表 E (表中项的个数设定为 10,用 a 到 j 十个英文字母表示),先随机生成 k 条

长度不同的根序列,并以这 k 条根序列代表 k 个簇,然后在每条根序列的基础上,通过随机变换序列长度和其中的项来合成其他全部序列。合成序列的特征通过如下的标识符进行描述: K 表示合成的数据集中簇的个数,C 表示数据集中序列的个数,L 表示最短根序列的长度, △表示其他根序列长度在最短根序列长度的基础上的递增度,VL 表示根序列所代表的簇中的其他序列长度相对于根序列变化的百分率约束,VP 表示根序列所代表的簇中的各序列变化的元素相对于根序列变化百分率的约束。例如,对于数据集 K6C6000L50△50VL5VP10 的解释就是,该数据集中有 6 个簇,一共 6 000 条序列分布在这 6 个簇中,因为最短根序列的长度为 50,所以其他根序列的长度在其基础上以 50 位单位递增,所以 6 条根序列的长度分别为 50,100,150,200,250,300。每个根序列 S 所代表的簇中,其他序列都是通过变化 S 长度(随机在任意位置上添加或删除项)的至多 5% 以及改变项(随机将序列中的任意项替换成其他项)的至多 10% 而合成的。

4.2　聚类效率的比较

对这两种算法进行测试的四组合成数据集特征如表 1 所示。对于这四组数据集,控制住其他特征不变,将参数 C 由 1 000 增长到 10 000。

表 1　合成的四组序列数据集

DatasetName	DatasetDescription
Dataste1	K5C1000L100△15VL10VP10
Dataset2	K5C4000L100△15VL10VP10
Dataset3	K5C7000L100△15VL10VP10
Dataset4	K5C10000L100△15VL10VP10

PSClu 与 EDClu 算法执行的时间对比见图 3。

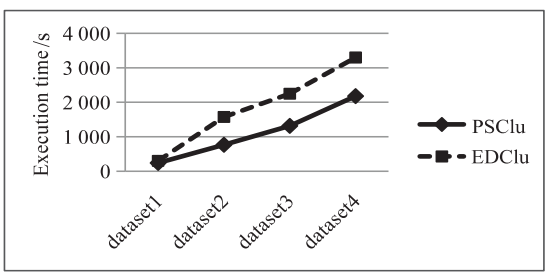

图 3　EDCluster 与 PSClu 聚类时间对比

从图中可以看出,带剪枝策略、使用序列局部相似性来代替全局相似性的 PSClu 算法的运行时间比不带剪枝策略的 EDClu 要少。

5　结束语

针对序列聚类算法中序列相似性度量算法的不足,提出了 PSClu 序列聚类算法。通过编辑距离的上下界以及前缀子序列的相似性计算减少序列之间两两编辑距离的计算。实验表明,PSClu 算法在时间效率上明显优于 EDClu 算法。由于在最差的情况下,PSClu 的时间复杂度仍然是与序列均长 L 呈平方项时间复杂度的,今后的研究目标将采用随机近似的方法进一步降低 PSClu 算法的时间复杂度。

参考文献:

[1] 戴东波,汤春蕾,熊　赟. 基于整体和局部相似性的序列聚类算法[J]. 软件学报,2010,21(4):702-717.

[2] 唐东明,朱清新,杨　凡,等. 一种有效的蛋白质序列聚类分析方法[J]. 软件学报,2011,22(8):1827-1837.

[3] WAGNER R A,FISCHER M J. The string-to-string correction problem[J]. Journal of the ACM,1974,21(1):168-173.

[4] 袁　方,周志勇,宋　鑫. 初始聚类中心优化的 k-means 算法[J]. 计算机工程,2007,33(3):65-66.

[5] 王　勇,唐　靖,饶勤菲,等. 高效率的 K-means 最佳聚类数确定算法[J]. 计算机应用,2014,34(5):1331-1335.

[6] CELEBI M E,KINGRAVI H A,VELA P A. A comparative study of efficient initialization methods for the k-means clustering algorithm[J]. Expert Systems with Applications,2013,40(1):200-210.

[7] HAMERLY G,ELKAN C. Alternatives to the k-means algorithm that find better clusterings[C]//Proceedings of the eleventh international conference on information and knowledge management. New York,NY,USA:ACM,2002:600-607.

[8] 戴　红,常子冠,于　宁. 数据挖掘导论[M]. 北京:清华大学出版社,2015.

[9] ILIOPOULOS C S,RAHMAN M S. New efficient algorithms for the LCS and constrained LCS problems[J]. Information Processing Letters,2008,106(1):13-18.

[10] SAKOE H,CHIBA S. Dynamic programming algorithm optimization for spoken word recognition[M]. [s. l.]:Morgan Kaufmann Publishers Inc.,1990.

[11] BANG K S,LU H,SIAU K. An efficient index structure for spatial databases[J]. Journal of Database Management,1996,7(3):3-16.

[12] RISTAD E S,YIANILOS P N. Learning string-edit distance[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence,1998,20(5):522-532.

[13] 姜　华,韩安琪,王美佳,等. 基于改进编辑距离的字符串相似度求解算法[J]. 计算机工程,2014,40(1):222-227.

[14] 刘莹霞. 链码技术和聚类分析在基因序列中的应用[D]. 广州:华南理工大学,2012.

[15] AGRAWAL R, SRIKANT R. Mining sequential patterns[C]//Eleventh international conference on data engineering. Washington, DC, USA: IEEE Computer Society, 1995: 3-14.