DOI: 10.11992/tis.201702019
网络出版地址: http://kns.cnki.net/kcms/detail/23.1538.tp.20180417.1130.008.html

去冗余 Top-k 对比序列模式挖掘

江冰, 谷飞洋, 何增有
(大连理工大学 软件学院, 辽宁 大连 116621)

摘　要: 对比序列模式可以用来表征不同类别数据集之间的差异。在生物信息、物流管理、电子商务等领域, 对比序列模式有着广泛的应用。Top-k 对比序列模式挖掘的目标是发现数据集中对比度最高的前 k 个序列模式。在 Top-k 对比序列模式挖掘中, 可能挖掘出冗余的序列模式。目前, 虽然有 Top-k 对比序列模式发现算法被提出, 但这些算法并未考虑冗余序列模式的问题。为此, 本文提出了基于广度优先生成树的去冗余 Top-k 对比序列模式挖掘算法 BFM(breadth-first miner)。使用 BFM 算法可以有效地解决冗余问题, 得到去冗余的 Top-k 对比序列模式。在 BFM 算法的基础上, 提出了性能更好的算法 PBFM(pruning breadth-first miner)。通过在真实数据集上的实验分析与对比 , 验证了本文算法的有效性。

关键词: 对比序列模式; 广度优先; 冗余序列模式; 模式挖掘; Top-k
中图分类号: TP393　　**文献标志码**: A　　**文章编号**: 1673−4785(2018)05−0680−07

中文引用格式: 江冰, 谷飞洋, 何增有. 去冗余 Top-k 对比序列模式挖掘[J]. 智能系统学报, 2018, 13(5): 680–686.
英文引用格式: JIANG Bing, GU Feiyang, HE Zengyou. Mining Top-k non-redundant distinguishing sequential patterns[J]. CAAI transactions on intelligent systems, 2018, 13(5): 680–686.

Mining Top-k non-redundant distinguishing sequential patterns

JIANG Bing, GU Feiyang, HE Zengyou
(Software School, Dalian University of Technology, Dalian 116621, China)

Abstract: Distinct sequential patterns can be used to characterize different categories of datasets. In the field of bioinformatics, logistics management, and e-commerce, the comparison of sequential pattern has a wide range of applications. The goal of the Top-k distinguishing sequential pattern mining is to find k patterns with the highest contrast in a given data set. However, in the Top-k distinguishing sequential pattern mining, there is a redundancy problem with respect to the set of reported sequential patterns, which is not considered by the algorithm. Therefore, in this paper, a non-redundant Top-k distinguishing sequential pattern mining algorithm, breadth-first miner (BFM), is proposed based on breadth-first spanning tree. The redundancy problem is effectively solved using the BFM algorithm. Based on the BFM algorithm, a better algorithm, pruning breadth-first miner (PBFM), is proposed. Through the experimental analysis and comparison on the real data set, the validity of the algorithm is verified.

Keywords: distinguishing sequential pattern; breadth-first; redundant sequential patterns; pattern mining; Top-k

至今已经有很多种序列模式被提出, 包括周期模式[1]、偏序模式[2]、闭合模式[3]、对比序列模式[4]、频繁序列模式[5]等。对比序列模式挖掘作为数据挖掘中重要的一个问题, 目前已经积累了大量的研究成果[6-7]。对比序列模式是指在一类数据集中频繁出现, 而在另一类数据集中很少出现的序列模式。对比序列模式可以描述不同数据集之间的差异, 在很多领域有广泛应用。例如, 对比序列模式可以用于纳税人行为分析[8], 患者的风险预测[9]等。在对比序列模式挖掘中, Top-k 对比序列模式挖掘是一种重要的挖掘策略。 Top-k 方法是指在给定的标准下挖掘出差异最大的 k 个模式的方法。该方法被广泛应用在关联规则[10]、

收稿日期: 2017−02−26.　**网络出版日期**: 2018−04−18.
基金项目: 国家自然科学基金项目 (61572094); 大学生创新创业训练计划项目 (201710141090101038 2).
通信作者: 何增有. E-mail: zyhe@dlut.edu.cn.

序列规则[11]、相关模式[12]和序列模式[7]等领域中。但是, 在 Top-k 策略挖掘结果中依然存在冗余的问题。针对这一问题, 本文提出了挖掘去冗余 Top-k 对比序列模式集合的方法。

1　相关工作

对比序列模式挖掘主要包括基于阈值的对比序列模式挖掘和 Top-k 对比序列模式挖掘两个研究方向。基于阈值的对比序列模式挖掘的目标是找出所有满足给定阈值的模式。最直接挖掘对比序列模式的方法是枚举所有的序列模式, 然后统计它们在每个类别上的频率。很明显这种方法的效率太低, 不能满足实际应用的需求。Chan 等[13]于 2003 年提出一种基于后缀树的挖掘对比序列模式的算法 (emerging substrings mining, ESM)。与朴素的挖掘算法相比, ESM 提高了一定的效率。Ji 等[14]于 2007 年定义了 MDS (minimal distinguishing subsequence) 的概念, 并提出了相应的挖掘算法, 即 ConSGapMiner (contrast sequences with gap miner) 算法。ConSGapMiner 是比较经典的对比序列模式挖掘算法, 它能以较快的效率挖掘出对比序列模式。但是 MDS 定义的对比序列模式在正例集中大于一个固定的阈值, 在反例集中小于一个固定阈值, 这种定义可能使一些有意义的模式没有被挖掘出来。2010 年 Deng 等[15]在 ConSGapMiner 的基础上利用 F-ratio 作为对比度; 2011 年 Yu 等[16]提出了 TSEPsMiner 算法; TSEPsMiner 利用 GrowthRate 作为对比度, 而且将这个对比度应用在了挖掘对比序列模式的算法中; 2014 年 Wang 等[17]提出用 gd-DSPMiner 算法来解决 MDS 定义中存在的问题。在不明确数据的差异程度时, 使用基于阈值的对比序列模式挖掘难以挖掘出预期的序列模式。在这种情况下, Top-k 对比序列模式挖掘是一个可行的办法。Top-k 算法不用设定对比度的阈值, 只需要设置想要挖掘模式的数目。杨皓等[7]于 2015 年提出了新的 Top-k 挖掘算法, 即 kDSP-Miner(Top-k distinguishing sequential patterns with gap constraint miner) 算法。但是 kDSP-Miner并没有考虑冗余问题, kDSP-Miner 挖掘出的序列模式可能会有大量的冗余。

2　问题定义

对于一个给定的数据集D, 它由两部分组成, 分别是D_+和D_-。其中, D_+是正例集, D_-是反例集。数据集D由多个序列组成。对于每个序列S, 有$S=e_1,e_2,\cdots,e_n$。其中e_i称为组成序列S的元素。用len(S)来表示序列S的长度, 即S中包含元素的数目。用$S[i]$表示序列S第i个位置的元素e_i。由数据集 D 中所有的元素组成的集合称为字母表, 用符号Σ表示。对于给定的两个序列S和S', 若存在一组整数$1\leqslant k_1<k_2<\cdots<k_m\leqslant \text{len}(S)$, 使得对于任意的$i\in[1,\text{len}(S')]$, 都有$S'[i]=S[k_i]$, 则称$S$是$S'$的超序列, S'是S的子序列。

例 1　对于序列, $S=\{A,C,G,T,C,A\}$, $S'=\{C,G,T\}$, 存在一组整数$k_1=2$, $k_2=3$, $k_3=4$, 使得$S[k_1]=S'[1],S[k_2]=S'[2],S[k_3]=S'[3]$, 所以$S'$是$S$的子序列, 记作$S'\subseteq S$。给定数据集$D$, 序列$P$在数据集$D$中的支持度由式 (1) 来定义:

$$\text{Sup}(P,D)=\frac{|\{S\in D|P\subseteq S\}|}{|D|} \tag{1}$$

式中$|D|$表示数据集D中包含序列的个数。

定义 1　(对比度) 对于给定的数据集D, D由正例集D_+和反例集D_-组成。序列P的对比度定义为

$$\text{CR}(P,D_+,D_-)=\text{Sup}(P,D_+)-\text{Sup}(P,D_-)$$

例 2　对于表 1 中给出的 DNA 数据集, $|D_+|=6$, $|D_-|=4$。令序列$P=\{A,G,T\}$, 有$\text{Sup}(P,D_+)=1$, $\text{Sup}(P,D_-)=0.25$。$\text{CR}(P,D_+,D_-)=\text{Sup}(P,D_+)-\text{Sup},(P,D_-)$, CR $(P)=0.75$。对于一个给定的数据集D, 如果序列P 满足$\text{CR}(P,D_+,D_-)>0$, 则称P是一个对比序列模式。

表 1　含有两个类别的基因数据集
Table 1　A gene data set with two classes

序列	类别	序列	类别
C, A, G, T, A	D_+	C, A, A, G, T, A	D_+
C, A, G, T, G	D_+	A, A, C, T	D_-
A, G, A, G, T, C	D_+	A, T, G, C	D_-
T, T, A, A, G, T, A	D_+	C, A, G, T	D_-
T, A, G, T, A, C	D_+	T, A, A, T	D_-

定义 2　(Top-k 对比序列模式) 给定正例集D_+和反例集D_-, 在所有的对比序列模式中,对比度最大的前 k 个序列模式称为 Top-k 对比序列模式。

Top-k 对比序列模式挖掘的目标是找出给定数据集中对比度最大的 k 个序列模式。

例 3　在表 1 的数据集中, 令$k=5$,则找出的 Top-k 对比序列模式见表 2 所示。

定义 3　(冗余对比序列模式) 对于两个给定的对比序列模式P和P', 如果满足:

1) P'是的P子序列, 即$P'\subseteq P$;

2) $\mathrm{CR}(P', D_+, D_-) \geqslant \mathrm{CR}(P, D_+, D_-)$;

则称模式P是冗余对比序列模式。

表 2 的对比序列模式中，$\{G,T\} \subseteq \{G,T,A\}$，且 $\mathrm{CR}(\{G,T\}, D_+, D_-) \geqslant \mathrm{CR}(\{G,T,A\}, D_+, D_-)$，所以模式$\{G,T,A\}$是相对于$\{G,T\}$的冗余对比序列模式。

表 2　表 1 中基因数据集的 Top-5 对比序列模式

Table 2　Top-five discriminative sequential patterns from gene data set in table 1

Sequence	Sup (P, D_+)	Sup (P, D_-)	Sup (P, D_+, D_-)
{A, G}	1	0.25	0.75
{G, T}	1	0.25	0.75
{A, G, T}	1	0.25	0.75
{G, T, A}	0.67	0	0.67
{A, G, T, A}	0.67	0	0.67

定义 4　(去冗余 Top-k 对比序列模式) 集合L满足 Top-k 对比序列模式集合的要求，同时对于每个序列$r_a \in L$, 不存在$r_b \notin L \wedge r_b \subseteq r_a \wedge \mathrm{CR}(r_b, D_+, D_-) \geqslant \mathrm{CR}(r_a, D_+, D_-)$; 对于任意序列$r_a \in L$, $r_c \in L$, r_a不是相对于r_c的冗余对比序列模式。

例 4　在表 1 的数据集中，令$k=5$, 则找出的去冗余 Top-k 对比序列模式如表 3 所示。

表 3　表 1 中基因数据集的 Top-5 去冗余对比序列模式

Table 3　Top-five non-redundant distinguishing sequential patterns from gene data set in table 1

Sequence	Sup (P, D_+)	Sup (P, D_-)	Sup (P, D_+, D_-)
{A, G}	1	0.25	0.75
{G, T}	1	0.25	0.75
{T, A}	0.67	0.25	0.42
{C, A}	0.5	0.25	0.25
{T, C}	0.17	0	0.17

本文中常用的符号及定义总结在表 4 中。

表 4　符号及其含义

Table 4　Symbols and their meaning

符号	含义
D	数据集
D_+	正例集
D_-	反例集
len (S)	序列 S 的长度
$S[i]$	序列 S 第 i 个位置的元素 e_i
$S' \subseteq S$	S'是 S 的子序列
Sup (P, D)	序列 P 在数据集 D 中的支持度
CR (P, D_+, D_-)	序列 P 的对比度

3　算法设计

为了挖掘去冗余 Top-k 对比序列模式的集合，用本文提出的 BFM 和 PBFM 算法，来解决挖掘出的结果集合的冗余问题。BFM 和 PBFM 算法基于广度优先生成树的原理来寻找 Top-k 序列的集合，树的生成过程就是 Top-k 集合的更新过程。相比于使用深度优先的方法来生成树结构，使用广度优先的方法每次更新时不改变 Top-k 集合的大小，所以不会出现冗余的 Top-k 集合。

3.1　广度优先的生成树算法

1) 根据给定的数据集D, 生成字母表Σ。

2) 创建 Top-k 集合 L, 设置集合L的最小对比度阈值$\min \mathrm{TopkCR} = 0$。

3) 创建一个队列，将字母表中的每个元素放入队列中。

4) 对于队列的第一个元素，在其末尾分别与字母表中的每个元素连接，形成新的序列。

5) 计算每个新的序列 P 的支持度和对比度，如果$\mathrm{Sup}(P, D_+) \geqslant \min \mathrm{TopkCR}$，将 P 放入队列中，否则不放入队列中。

当$|L| < k$时，在集合 L 中寻找序列 P 的子序列，若未找到子序列，将 P 加入集合 L; 若找到了子序列 P, 且 P 相对于 P'不是冗余序列，则将 P 加入集合 L, 并更新集合L的最小对比度阈值$\min \mathrm{TopkCR}$ (若$\mathrm{CR}(P, D_+, D_-) \leqslant \min \mathrm{TopkCR}$, 则 $\min \mathrm{TopkCR} = \mathrm{CR}(P, D_+, D_-)$), 否则不加入集合。

当$|L| = k$时，如果$\mathrm{CR}(P, D_+, D_-) > \min \mathrm{TopkCR}$，在集合 L 中寻找序列 P 的子序列，若未找到子序列，用 P 替换集合 L 中对比度最小的序列; 若找到了子序列 P', 且 P 相对 P'不是冗余序列，则用 P 替换集合 L 中对比度最小的序列，并更新集合 L 的最小对比度阈值$\min \mathrm{TopkCR}$(若此时集合中第二小的 CR 小于 $\mathrm{CR}(P, D_+, D_-)$，则将它设置为 $\min \mathrm{TopkCR}$，否则$\min \mathrm{TopkCR} = \mathrm{CR}(P, D_+, D_-)$); 否则不替换。

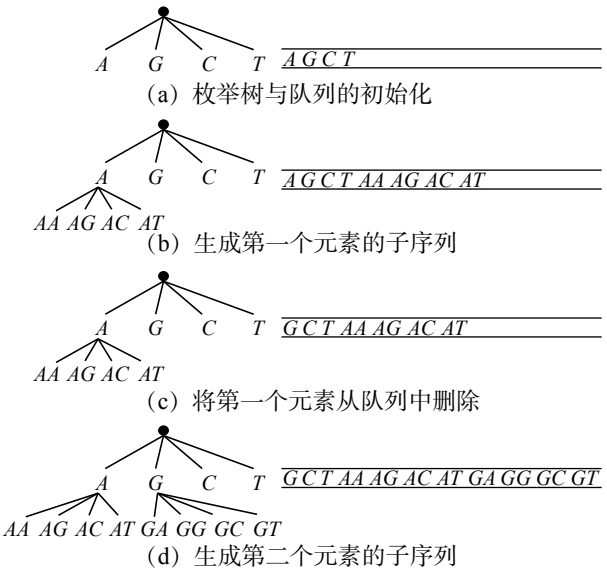
6) 将队列的第一个元素弹出。

7) 重复 4)~6), 直到队列为空。

例 5　对表 1 的数据集进行去冗余 Top-k 对比序列模式挖掘，令 k =5。找出基因数据集的字母表Σ为$\Sigma = \{A, G, C, T\}$。将字母表的每个元素放入队列中，生成的树结构和队列如图 1 所示。

在 Top-k 对比序列模式挖掘中，去除冗余序列模式是提高挖掘结果质量的重要一步。但是在原有挖掘过程的基础上，加入去冗余的步骤后，一个新的序列可能会替换 Top-k 集合中的多个序

列, 使 Top-k 集合中的序列模式数目小于 k 个。针对以上问题, 本文提出基于广度优先的生成树算法 BFM (breadth-first miner) 来去除冗余的序列模式。使用 BFM 算法可以在去除冗余序列模式的同时, 保证 Top-k 集合的大小不发生变化。



(a) 枚举树与队列的初始化
(b) 生成第一个元素的子序列
(c) 将第一个元素从队列中删除
(d) 生成第二个元素的子序列

图 1　生成树和队列的动态变化

Fig. 1　The dynamic change of spanning tree and queue

算法 1　BFM (breadth-first miner)

输入　正例集D_+、反例集D_-、参数k。

输出　包含 Top-k 对比序列模式的集合L。

/*初始化*/

1)创建集合L, 设置集合L的最小对比度阈值 $\min\text{TopkCR}=0$;

2)创建队列, 初始化队列 queue 为空;

3)计算出字母表Σ, 将字母表中的每个元素加入到 queue 中;

4)创建树的根节点,建立由根节点分别指向字母表中每个元素的连接, 令 min TopkCR = get-MinTopkCR (L);

5)对Σ中的每一个元素e, 把e添加到 queue 的第一个序列的末尾, 组成新的序列 P, 如果 $\text{Sup}(P,D_+)\geqslant\min\text{TopkCR}$ 则把 P 加入到队列中;

如果 $|L|<k$ 则在集合L寻找中P的子序列P';

如果 P'被找到且 P 不是相对于 P'的冗余序列模式, 则把 P 加入集合 L, 更新$\min\text{TopkCR}$;

否则把 P 加入集合 L, 更新$\min\text{TopkCR}$;

6)如果 $|L|=k$, 并且$\text{CR}(P,D_+,D_-)>\min\text{TopkCR}$ 那么在集合 L 中寻找 P 的子序列 P';

如果 P'被找到并且 P 是相对于 P'的冗余序列, 则用 P 替换集合 L 中对比度最小的序列更新 $\min\text{TopkCR}$, 否则, 用 P 替换集合 L 中对比度最小的序列更新$\min\text{TopkCR}$;

7)重复步骤 5)、6), 直到对列为空。

3.2　剪枝策略

为了提高算法的性能, 本文中应用了一系列剪枝策略来辅助算法的运行[7]。运用这些剪枝策略, 可以使程序运行的效率提高, 更快地找出 Top-k 集合。

剪枝策略 1　(反例元素剪枝策略) 在遍历数据集D生成字母表Σ时, 只统计正例集D_+中出现的元素, 而不统计反例集D_-中出现的元素。

这条剪枝策略基于以下原理: 如果元素e不在正例集D_+中, 则所有包含元素e的超序列 E 也不在正例集中, $\text{Sup}(E,D_+)=0$。由对比序列模式的定义可知, 对比序列模式必须满足$\text{CR}(P,D_+,D_-)>0$, 即 $\text{Sup}(P,D_+)-\text{Sup}(P,D_-)>0$。因为$\text{Sup}(E,D_+)=0$, $\text{Sup}(E,D_-)\geqslant 0$, 所以序列 E 不满足$\text{Sup}(P,D_+)-\text{Sup}(P,D_-)>0$, E 不是对比序列模式, 字母表Σ中不需要包含元素e。

剪枝策略 2　(序列支持度的剪枝策略) 当 $|L|=k$时, 如果序列P满足$\text{Sup}(P,D_+)\leqslant\min\text{TopkCR}$, 则不把序列$P$放入队列。

由 $\text{CR}(P,D_+,D_-)=\text{Sup}(P,D_+)-\text{Sup}(P,D_-)$ 可知, $\text{Sup}(P,D_-)\geqslant 0$, 如果$\text{Sup}(P,D_+)\leqslant\min\text{TopkCR}$ 则 $\text{CR}(P,D_+,D_-)\leqslant\min\text{TopkCR}$, 序列 P 不是 Top-k 对比序列模式。对于任意一个模式 P 的超模式P', 有 $\text{Sup}(P',D_+)\leqslant\text{Sup}(P,D_+)$, 因此$\text{Sup}(P',D_+)\leqslant\min\text{TopkCR}$ 也成立。所以序列 P'也不是 Top-k 对比序列模式。序列 P 不用放入队列, 即把以序列 P 为根节点的子树从整体的生成树上剪枝。

剪枝策略 3　(元素支持度的剪枝策略) 当 $|L|=k$时, 如果元素e满足$\text{Sup}(e,D_+)\leqslant\min\text{TopkCR}$, 则将元素$e$从字母表中移除。

由 Top-k 对比序列模式的定义可知, 包含元素e的序列不是 Top-k 对比序列, 所以在生成树结构时, 不用生成包含元素e的序列, 可以将元素e从字母表移除。

加入以上 3 条剪枝策略后, 树结构生成的速度会加快, 可以在更短的时间内找到 Top-k 对比序列模式。对于某一类数据集, 使用剪枝后, 算法的效率会显著提升。加入剪枝后的算法如算法 2。

算法 2　PBFM(pruning breadth-first miner)

输入　正例集D_+, 反例集D_-, 参数k。

输出　包含 Top-k 对比序列模式的集合 L。

/*初始化*/

1）创建集合 L，设置集合 L 的最小对比度阈值 $\min \text{TopkCR} = 0$；

2）创建队列，初始化队列 queue 为空；

3）计算正例集 D_+ 的字母表 Σ，将字母表中的元素加入到 queue 中；

4）创建树的根节点，建立由根节点分别指向字母表中每个元素的指针，令 min TopkCR = getMinTopkCR (L)；

5）如果 $(\text{Sup}(e, D_+) \leqslant \min \text{TopkCR})$ 则从字母表 Σ 中删除元素 e，对 Σ 中的每一个元素 e，把 e 添加到队列的第一个序列的末尾，组成新的序列 P；

如果 $\text{Sup}(P, D_+) \geqslant \min \text{TopkCR}$，则把 P 加入到队列中；

如果 $|L| < k$，则在集合 L 寻找 P 的子序列 P'；

如果 P' 被找到且 P 不是相对于 P' 的冗余序列，则把 P 加入集合 L，更新 $\min \text{TopkCR}$，否则把 P 加入集合 L，更新 min TopkCR；

6）如果 $|L| = k$，并且 $\text{CR}(P, D_+, D_-) > \min \text{TopkCR}$，则在集合 L 寻找 P 的子序列 P'；

如果 P' 被找到，并且 P 不是相对于 P' 的冗余序列，则用 P 替换集合 L 中对比度最小的序列，更新 $\min \text{TopkCR}$，否则用 P 替换集合 L 中对比度最小的序列更新 $\min \text{TopkCR}$；

7）重复步骤 5）、6），直到队列为空。

4　实验结果

4.1　实验环境

本文设计了一系列实验来评估算法的有效性。算法用 C++语言来实现。实验中所用到的数据集为 4 组真实数据。这 4 组数据分别是：E.Coli 数据集，记录了两个不同类型的 DNA 序列。在 E.Coli 数据集中，每个 DNA 序列前面都用“+”和“–”标记出了序列所属的类别。UJI 数据集，记录了超过 11 000 个独立的手写数字。ADLs 数据集，记录了一段时间内不同的人在自己家中的活动情况。Question 数据集，记录了各种不同的问题，可以将每个问题看作由不同单词组成的序列。前 3 个数据集来自 UCI 的机器学习数据集。最后一个数据集是 Question 的训练数据集。实验运行的环境是：Core i3 的处理器，Windows 7 操作系统，2GB RAM 的计算机上完成。表 5 中列出了实验中用到的数据集的特征。

表 5　数据集的特征

Table 5　The characteristics of the data sets

数据集	类别	$\lvert D_+\rvert$	$\lvert D_-\rvert$	$\lvert\Sigma\rvert$	$\lvert D\rvert$
E.Coli	E.Coli+E.Coli-	53	53	4	106
UJI	Write+Write-	784	784	10	1 568
ADLs	Activity+Activity-	13	21	9	34
Question	Question+Question-	151	156	146	307

4.2　实验结果分析

1) 实验 1(去冗余前后 Top-k 集合对比实验)

实验 1 的目标是比较去冗余前后 Top-k 集合中序列模式的变化，来验证去冗余算法的有效性。在该实验中，使用了 3 组数据来对比去冗余前后 Top-k 结果集合的不同。每组数据分别找出了含有冗余序列的 Top-k 集合和不含冗余序列的 Top-k 集合，并比较其中序列的组成。在实验中，k 值设置为 5。实验结果如表 6~8 所示。

表 6　去冗余前后 Top-k 序列模式集合的变化 (ADLs 数据集)

Table 6　The set of Top-k sequential patterns before and after eliminating redundancy(ADLs data set)

冗余 Top-k 集合		去冗余 Top-k 集合	
序列	对比度	序列	对比度
洗澡、吃早餐	1	洗澡、吃早餐	1
梳妆、洗澡、吃早餐	0.846 154	梳妆、洗澡、吃早餐	0.769 231
上厕所、梳妆、洗澡	0.769 231	睡觉、上厕所、梳妆、洗澡	0.692 308
洗澡、吃早餐、休息	0.769 231	睡觉、上厕所、梳妆	0.597 07
上厕所、梳妆、洗澡、吃早餐	0.769 231	梳妆、洗澡	0.560 44

通过实验结果可以发现，去冗余 Top-k 集合中出现了冗余 Top-k 集合中没有出现的序列模式。同时，去冗余 Top-k 集合中删去了冗余的序列模式。因此，本文的算法能够有成效地去除冗余对比序列模式。

2) 实验 2(加入剪枝策略前后的对比实验)

实验 2 的目标是比较算法 1 与加入剪枝策略后算法效率的变化。分别在 ADLs 数据集和 Question 数据集上进行对比实验，比较算法 1 和算法 2 运行的时间，来衡量算法的效率。

表 7　去冗余前后 Top-k 序列模式集合的变化 (E.Coli 数据集)

Table 7　The set of Top-k sequential patterns before and after eliminating redundancy(E.Coli data set)

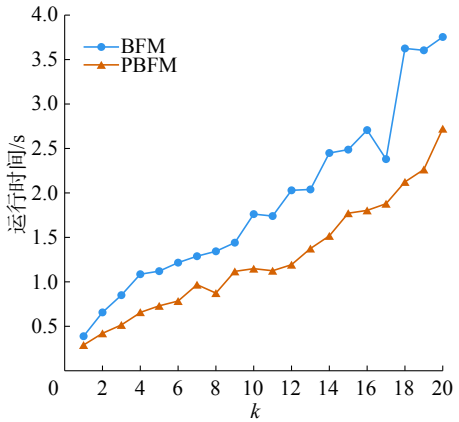
冗余 Top-k 集合		去冗余 Top-k 集合	
序列	对比度	序列	对比度
ATA	0.396 226	ATA	0.396 226
AAA	0.452 830	AAA	0.452 830
TATA	0.396 226	TTTA	0.377 358
AATT	0.415 094	TATA	0.396 226
AAAA	0.415 094	AATT	0.415 094

表 8　去冗余前后 Top-k 序列模式集合的变化 (UJI 数据集)

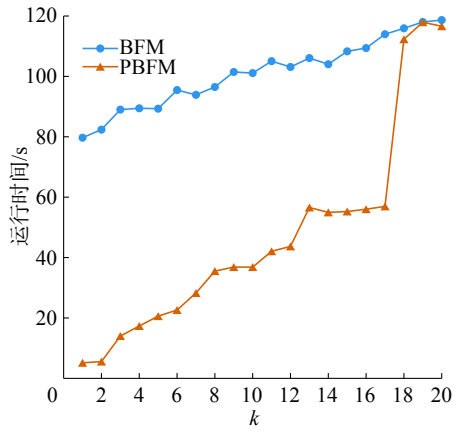
Table 8　The set of Top-k sequential patterns before and after eliminating redundancy(UJI data set)

冗余 Top-k 集合		去冗余 Top-k 集合	
序列	对比度	序列	对比度
712	0.081 632 7	712	0.081 632 7
317	0.079 081 6	317	0.079 081 6
931	0.088 010 2	931	0.088 010 2
610	0.088 010 2	610	0.088 010 2
126	0.079 081 6	126	0.079 081 6

将 k 的值分别从 1 取到 20, 比较算法 1(BFM) 和算法 2(PBFM) 运行的时间如图 2 所示。从图 2 中可以看出, 随着 k 值的增加, 两种算法的运行时间都在增长, 但 PBFM 的运行时间明显少于 BFM 的运行时间。在 ADLs 数据集中, 随着 k 值逐渐变大, 这一区别越来越明显。在 Question 数据集中, 当 k 值较小时, 这一区别较为明显。随着 k 值的增大, Top-k 集合的最小对比度 minTopkCR 逐渐变小, 当$k \geqslant 18$时, PBFM 算法中删除的元素个数较少, 但 PBFM 算法运行的时间仍然少于 BFM 算法运行的时间。



(a) ADLs数据集



(b) Question数据集

图 2　BFM 和 PBFM 的效率对比

Fig. 2　Comparison of BFM and PBFM efficiencies

5　结束语

本文首先提出了一种挖掘去冗余 Top-k 对比序列模式的算法 BFM,这是一种基于广度优先生成树的算法。通过不断的比较子序列和超序列的对比度, Top-k 集合不断地更新, 直到树结构的生成过程结束。相比之前的 Top-k 对比序列模式挖掘算法, BFM 算法可以得到去冗余的 Top-k 集合, 并且不需要其他集合的辅助。

在 BFM 算法的基础上,提出了性能更好的 PBFM 算法。与 BFM 算法相比, PBFM 算法可以在更短的时间内完成挖掘任务, 并且不需要额外的操作。

参考文献:

[1] ZHANG Minghua, KAO Ben, CHEUNG D W, et al. Mining periodic patterns with gap requirement from sequences [J]. ACM transactions on knowledge discovery from data, 2007, 1(2): 7.

[2] PEI Jian, WANG Haixun, LIU Jian, et al. Discovering frequent closed partial orders from strings[J]. IEEE transactions on knowledge and data engineering, 2006, 18(11): 1467–1481.

[3] YAN Xifeng, HAN Jiawei, AFSHAR R. CloSpan: mining: closed sequential patterns in large datasets[C]//Proceedings of the 3rd SIAM International Conference on Data Mining. San Francisco, USA, 2003: 166–177.

[4] JI Xiaonan, BAILEY J, DONG Guozhu. Mining minimal distinguishing subsequence patterns with gap constraints [J]. Knowledge and information systems, 2007, 11(3): 259–286.

[5] ZAKI M J. SPADE: an efficient algorithm for mining frequent sequences[J]. Machine learning, 2001, 42(1/2):

31–60.

[6] YANG Hao, DUAN Lei, DONG Guozhu, et al. Mining itemset-based distinguishing sequential patterns with gap constraint[C]//Proceedings of the 20th International Conference on Database Systems for Advanced Applications. Hanoi, Vietnam, 2015: 39–54.

[7] 杨皓, 段磊, 胡斌, 等. 带间隔约束的 Top-k 对比序列模式挖掘[J]. 软件学报, 2015, 26(11): 2994–3009.
YANG Hao, DUAN Lei, HU Bin, et al. Mining Top-k distinguishing sequential patterns with gap constraint[J]. Journal of software, 2015, 26(11): 2994–3009.

[8] ZHENG Zhigang, WEI Wei, LIU Chunming, et al. An effective contrast sequential pattern mining approach to taxpayer behavior analysis[J]. World wide web, 2016, 19(4): 633–651.

[9] GHOSH S, FENG Mengling, NGUYEN H, et al. Risk prediction for acute hypotensive patients by using gap constrained sequential contrast patterns[J]. AMIA annual symposium proceedings, 2014, 2014: 1748–1757.

[10] FOURNIER-VIGER P, TSENG VS. Mining Top-K Non-redundant association rules[C]//Proceedings of the 20th International Symposium on Foundations of Intelligent Systems. Macau, China, 2012, 7661: 31–40.

[11] FOURNIER-VIGER P, TSENG V S. TNS: mining top-k non-redundant sequential rules[C]//Proceedings of the 28th Symposium on Applied Computing. Coimbra, Portugal, 2013: 164–166.

[12] KAMEYA Y, SATO T. RP-growth: Top-k mining of relevant patterns with minimum support raising[C]//Proceedings of the 12th SIAM International Conference on Data Mining. Anaheim, USA, 2012: 816–827.

[13] CHAN S, KAO B, YIP C L, et al. Mining emerging substrings[C]//Proceedings of 8th International Conference on Database Systems for Advanced Applications. Kyoto, Japan, 2003: 119–126.

[14] JI Xiaonan, BAILEY J, DONG Guozhu. Mining minimal distinguishing subsequence patterns with gap constraints [J]. Knowledge and information systems, 2007, 11(3): 259–286.

[15] DENG Kang, ZAIANE O R. An occurrence based approach to mine emerging sequences[C]//Proceedings of the 12th International Conference on Data Warehousing and Knowledge Discovery. Bilbao, Spain, 2010: 275–284.

[16] YU H H, CHEN Chunhao, TSENG V S. Mining emerging patterns from time series data with time gap constraint[J]. International journal of innovative computing information and control, 2011, 7(9): 5515–5528.

[17] WANG Xianming, DUAN Lei, DONG Guozhu, et al. Efficient mining of density-aware distinguishing sequential patterns with gap constraints[C]//Proceedings of the 19th International Conference on Database Systems for Advanced Applications. Bali, Indonesia, 2014: 372–387.

作者简介：

江冰，女，1995 年生，硕士研究生，主要研究方向为数据挖掘和生物信息。

谷飞洋，男，1991 年生，硕士研究生，主要研究方向为数据挖掘和生物信息。

何增有，男，1976 年生，教授，博士生导师，博士，主要研究方向为数据挖掘、生物信息。获得省部级奖励 3 项。发表国际期刊论文 40 余篇，据 Google Scholar 统计，发表的论文被引用 2600 余次，出版英文学术专著 1 部。