·人工智能与模式识别· 文章编号：1000-3428(2021)08-0045-09 文献标志码：A 中图分类号：TP391

面向对比序列模式发现的独立精确置换检验算法

吴 军，欧阳艾嘉，张 琳
（遵义师范学院 信息工程学院，贵州 遵义 563000）

摘 要：传统的对比序列模式挖掘算法存在一定数量的假阳性对比序列模式，其提供的错误信息会干扰后续任务的决策。设计一种IEP-DSP算法过滤假阳性对比序列模式。运用spade方法和WRAcc对比性度量找到候选对比序列模式和所有置换数据集合中的对比序列模式，通过模拟置换过程，使用独立精确置换检验方法为不同长度的模式建立独立精确零分布，并计算每个候选对比序列模式的精确p-value，运用错误发现率度量将各个长度的假阳性对比序列模式数量控制在置信度为α的统计显著水平下。在真实数据集和仿真数据集上的实验结果表明，IEP-DSP算法够过滤掉大量的假阳性对比序列模式，相比基于统计显著性检验的方法能保留更多的真对比序列模式，验证了独立精确置换检验相较于标准置换检验的优越性。

关键词：数据挖掘；模式发现；对比序列模式挖掘；统计显著性检验；独立精确置换检验

开放科学（资源服务）标志码（OSID）：

中文引用格式：吴军，欧阳艾嘉，张琳.面向对比序列模式发现的独立精确置换检验算法[J].计算机工程，2021，47(8)：45-53，61.

英文引用格式：WU J，OUYANG A J，ZHANG L.Independent exact permutation testing algorithm for distinguishing sequential pattern discovery[J].Computer Engineering，2021，47(8)：45-53，61.

Independent Exact Permutation Testing Algorithm for Distinguishing Sequential Pattern Discovery

WU Jun，OUYANG Aijia，ZHANG Lin
（School of Information Engineering，Zunyi Normal University，Zunyi，Guizhou 563000，China）

【Abstract】Traditional distinguishing sequential pattern mining algorithms usually generate a number of false positive patterns in their results，which hinder the subsequent decisions of tasks. To address the problem，a method named IEP-DSP for filtering out false positive patterns is proposed.The method employs the spade algorithm and the WRAcc measure to produce the distinguishing sequential patterns to be tested and the distinguishing sequential patterns that exist in permutated sequential data sets.Through the simulated permutation process，the independent exact permutation testing method is used to establish independent exact null distributions for patterns with different length，and the exact p-value of the tested patterns can be calculated from these null distributions.The False Discovery Rate（FDR）measure is used to control the number of false positive distinguishing patterns with different length under a confidence level α. The experimental results on real data sets and simulated data sets show that the IEP-DSP algorithm can eliminate a large number of false positive distinguishing patterns while keeping more real distinguishing sequential patterns.At the same time，the advantage of independent exact permutation testing over standard permutation testing is proved.

【Key words】data mining；pattern discovery；distinguishing sequential pattern mining；statistical significance testing；independent exact permutation testing

DOI：10. 19678/j. issn. 1000-3428. 0058601

0 概述

在现实世界的许多应用中都存在大量的序列数据，如基因序列、文本序列、轨迹序列等。发现序列数据中的序列模式是一个十分重要的研究问题[1-2]。其中，在不同类型的序列数据分布中呈现显著对比

基金项目：国家自然科学基金（61662090）；贵州省教育厅青年科技人才成长项目（黔教合KY字[2017]250）；贵州省科技厅联合基金（黔科合LH字[2017]7069）；贵州省教育厅工程研究中心项目（黔教合KY字[2016]018）。

作者简介：吴 军（1990—），男，讲师、硕士，主研方向为数据挖掘、深度学习、生物信息学；欧阳艾嘉，教授、博士；张 琳，副教授、硕士。

收稿日期：2020-06-10 **修回日期：**2020-07-14 **E-mail：**wujun.myway@gmail.com

性的模式被称作对比序列模式[3]。对比序列模式具有非常重要的应用价值,比如在生物蛋白质序列中发现生物标记[4]、在风险评估和管理中预防攻击行为[5]等。

为了挖掘对比序列模式,一些方法被相继提出[3,6-8]。这些方法将注意力主要集中在对比性度量选择以及阈值约束设定上,使得结果中会存在一定数量偶然满足了算法约束但不能体现真实对比性的对比序列模式。这样的模式被称为假阳性模式,它们提供的错误信息会对后续分析产生严重的干扰。

DSPM-MTC方法运用统计显著性检验过滤了结果中的部分假阳性对比序列模式[9],其使用直接计算法来计算p-value值。在统计显著性检验中,每个被检验的对比序列模式会根据其分布信息计算得到一个p-value值,该值的大小度量了其统计显著性。对比序列模式的p-value值越小,则为假阳性模式的可能性就越小。

标准置换检验是一种常用的统计显著性检验方法,在非序列数据的模式发现任务中其检验效力高于直接计算法[10]。标准置换检验通过置换数据类型标签生成一定数量的置换数据集合,从中计算得到对比性度量值并建立相应的零分布,从而由该零分布计算得出被检验的对比序列模式的p-value值。值得注意的是,标准置换检验通常只执行一定次数的置换过程,因此其生成的只是精确零分布的一个近似零分布。使用该近似零分布检验挖掘结果存在p-value值可能为0、零分布共享、结果不唯一和计算开销大4个缺点,这些缺点限制了标准置换检验的实用性。

经过分析发现,导致标准置换检验上述缺点的原因是其构建的零分布是一个共享近似零分布。为此,本文提出一种通过模拟置换过程构建独立精确零分布的解决方案。通过设计基于独立精确置换检验的IEP-DSP算法,挖掘统计显著的对比序列模式,找到原始数据集合中和置换数据集合中的对比序列模式,并根据长度进行分组,计算置换数据集合每组中各个模式的对比性度量值分布,合并置换数据集合每组中的对比性度量值分布构建各自的独立精确零分布,通过独立精确零分布计算原始数据集合每组中候选对比序列模式的精确p-value值,并运用错误发现率(False Discovery Rate,FDR)度量将每组的假阳性模式数量约束在置信度为α的统计显著水平下,以保留更多的真对比序列模式。

1 相关工作

数据挖掘领域的目标是从数据中发现有价值的信息。为了得到正确信息,对数据挖掘算法结果进行评估成为当前热门研究问题[11-13]。在对比序列模式挖掘任务中,传统的挖掘算法将注意力放在了约束度量的设计和挖掘效率的优化上[3,6-8],没有对挖掘到的对比序列模式进行质量评估,即判别挖掘到的模式是否真实地体现了数据类别的特征。

运用统计显著性检验评估挖掘到的模式质量成为模式发现领域中热门研究方向,并相继提出一些不同策略的统计显著性检验方法。这些方法在模式挖掘过程中评估模式质量,或者在挖掘后的结果中进行模式质量评估。BRIN等[14]运用chi-square检验评估挖掘到统计显著性模式,然后根据一个设定的阈值过滤掉非统计显著的模式;ZHANG等[15]定义了一种新的模式SQ规则,并提出了一种随机检验的方法用于发现统计显著的SQ规则。WEBB[16]认为上述方法随着假设数量的增加,假阳性模式的数量也会增加,并针对该缺点,提出了直接计算法。LIU等[10]运用标准置换检验发现统计显著模式,并提出一次挖掘技术和预存储技术减少标准置换检验的计算开销;随后,研究人员提出2个改进的置换检验算法[17-18],这2个算法避开挖掘计算生成零分布,运用westfall-young置换过程计算得到模式的置换检验近似p-value,从而提升了置换检验用于模式发现任务的效率;PELLEGRINA等[19]设计了Spumante算法,该算法运用一种新颖的无条件检验找到统计显著的模式。无条件检验与Fisher检验等条件检验相比,对数据的假设要求更少。

以上方法仅在非序列数据的模式发现问题中得到了验证。为了提高序列数据中挖掘到的模式的质量,HE等[9]设计了DSPM-MTC算法挖掘统计显著的对比序列模式。该算法首先生成每个被检验模式的超几何分布,然后根据该分布直接计算得到模式的p-value值并进行非统计显著模式过滤,这种根据服从分布计算p-value值的方法称为直接计算法。文献[10]验证了在非序列数据集中,标准置换检验方法的性能优于直接计算法,但是由于置换的随机性,标准置换检验存在4个缺点。为探索置换检验对序列数据模式发现任务的有效性,并考虑到标准置换检验的缺点,本文提出使用独立精确置换检验的IEP-DSP算法挖掘统计显著的对比序列模式,以进一步提升报告的对比序列模式的质量。

2 问题描述

2.1 对比序列模式挖掘

令字母表为$E=\{e_1,e_2,\cdots,e_{|E|}\}$,一个序列模式$t$是由$E$中元素构成的一个有序符号列表$<m_1,m_2,\cdots,m_k>$,其中$m_i\in E$。如果一个序列模式$t$包含$k$个元素,则$t$的长度为$k$。给定2个序列模式$t_1=<m_1,m_2,\cdots,m_k>$

和 $t_2=<m^*_1, m^*_2, \cdots, m^*_{k*}>$,如果 t_2 的每一个元素 m^*_j 都存在于 t_1 中,且符合 t_1 的元素顺序,则 t_2 被称作是 t_1 的子序列,表示为 $t_2 \supseteq t_1$。给定一个包含 n 条序列的数据集合 $D=\{s_1, s_2, \cdots, s_n\}$ 和某个序列模式 t,t 在 D 中的支持度 $\sup(t, D)$ 被定义为 $|\{ s_i \mid t \supseteq s_i, s_i \in D\}|$,即 D 中包含 t 的序列数量。当且仅当序列模式 t 在 D 中的支持度超过了自定阈值 θ_{sup},t 就被认为是 D 中的频繁序列模式。目前,已经提出了许多频繁序列模式挖掘算法[20],如 GSP、Spade、PrefixSpan 等算法。

假设数据集合 D 含有 v 个类型标签,即 $D=\{D_1, D_2, \cdots, D_v\}$,若序列模式 t 在不同 D_i 中的支持度 $\sup(t, D_i)$ 呈现显著对比性,则 t 被称为对比序列模式。上述对比性可以由不同的对比性度量量化[21],例如 Growth rate、Diffsup、OddsRatio 等。为了便于阐明本文提出方法,后续讨论均假定 $D=\{D_1, D_2\}$。

对比序列模式挖掘任务的目标是找到所有支持度不小于 θ_{sup} 且对比度性度量值不小于 θ_{dis} 的序列模式,即频繁且存在对比性的序列模式。

2.2　标准置换检验

由于传统的对比序列模式挖掘算法只考虑了对比性度量约束,从而结果中会存在一定数量的假阳性模式,假阳性模式没有真正体现不同类型数据集的对比特征。统计显著性检验被广泛应用于假阳性结果的过滤,运用统计显著性检验进行质量评估时,建立的零假设为对比序列模式在 D_1 和 D_2 中具有相同的分布。同时,每个对比序列模式会被分配一个 p-value 值度量其统计显著性。一个对比序列模式 t 的 p-value 值的定义是在假设零假设为真的前提下,获得一个至少与 t 同样极端的对比序列模式的概率,这里的极端主要体现在对比性度量值的大小。

一般地,可以通过设定一个 p-value 值的置信度阈值 α 决定是否拒绝零假设,但当有多个对比序列模式需要被同时检验时,即多重假设检验,这种策略会导致假阳性结果的增加。FDR 是多重假设检验中常用的度量约束,其定义是整个结果中假阳性对比序列模式比例的期望值,可以使用 BH 方法约束整个结果的 FDR 值[22]。

标准置换检验是一种常用的统计显著性检验方法[10],其核心过程如图 1 所示。首先,挖掘原始数据集合 D_1 中的候选对比序列模式 R;然后,根据零假设生成一定数量的置换数据集合,挖掘并计算每个置换数据集合中对比序列模式的对比性度量值;最后,用所有计算得到的对比性度量值建立该置换检验的零分布,并通过该零分布计算所有候选对比序列模式的 p-value 值。

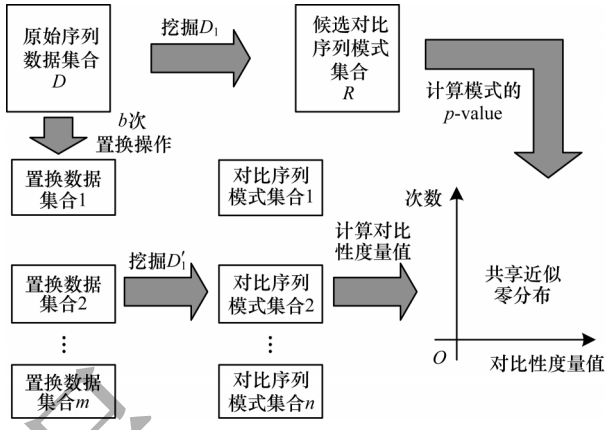


图 1　标准置换检验过程

Fig.1　Process of standard permutation testing

在标准置换检验中,穷举生成一个集合所有可能的置换数据集合是不切实际的,所以通常只产生一定数量的置换数据集合,这导致了构建的零分布不是精确零分布。使用该近似零分布进行模式统计显著性评估存在以下 4 个缺点:

1)某些对比序列模式的 p-value 值计算结果为 0;

2)共享同一个零分布会增大模式之间的互相影响;

3)同一数据集进行多次检验得到的统计显著的对比序列模式数量不一致;

4)增大置换次数会导致计算开销的显著增大。

这些缺点会大幅降低标准置换检验的实用性。分析发现造成标准置换检验 4 个缺点的根本原因是置换过程构建了一个共享近似零分布。因此,快速构建独立精确零分布是去除 4 个缺点的一个可行的解决方案。

3　IEP-DSP 算法

IEP-DSP 算法从序列数量分布出发,运用排列组合的思想模拟置换过程,直接计算得到不同长度对比序列模式的置换检验独立精确零分布。

3.1　候选对比序列模式

IEP-DSP 算法选定 WRAcc(Weighted Relative Accuracy)作为对比性度量[21]。给定一个对比序列模式 t,其 WRAcc 值主要考虑了 2 个部分信息:t 的相对支持度和 t 的支持度比率与数据比率的差别。具体的 WRAcc 值的计算公式为:

$$W_{wra}(t,q)=\frac{\sup(t,D)}{|D|}\left(\frac{q}{\sup(t,D)}-\frac{|D_1|}{|D|}\right) \quad (1)$$

其中:q 表示 D_1 中包含 t 的序列数量,即支持度 $\sup(t, D_1)$。

IEP-DSP 算法运用 Spade 算法挖掘频繁序列模式[23]。Spade 算法先将数据集中的序列表示为垂直结构,再运用序列联合操作构建树形结构以找到所

有的频繁序列模式。如果一个频繁序列模式的对比性度量值超过了阈值θ_{dis},则该频繁序列模式被称为候选对比序列模式,表示为t°。

3.2 独立精确置换检验

给定置换数据集合中的一个对比序列模式t',数据置换过程会改变它在置换数据集合D'_1和D'_2中的序列数量分布。假设t'在D'_1中的支持度为q',则它在D'_1和D'_2中的序列数量分布如表1所示。

表1 模式t的序列数量分布

Table 1 Sequence number distribution of pattern t

序列数量	D'_1	D'_2	总计
包含t	q'	$\sup(t',D)-q'$	$\sup(t',D)$
不包含t	$\lvert D_1\rvert-q'$	$\lvert D_2\rvert-\sup(t',D)+q'$	$\lvert D\rvert-\sup(t',D)$
总计	$\lvert D_1\rvert$	$\lvert D_2\rvert$	$\lvert D\rvert$

从表1可以看出,给定q'值后其余数值均可以写成基于q'的计算公式,即对于一个确定的q',t'在D'_1和D'_2的数量分布是唯一的。

独立精确置换检验的过程如图2所示。首先,找到候选对比序列模式R和所有可能在置换数据集合中出现的对比序列模式R',并根据模式长度进行各自分组;其次,针对R'_k集合中每个对比序列模式t',计算出其相应的对比性度量值分布;再次,合并R'_k集合中每个对比序列模式t'的对比性度量值分布即得到R'_k对应的独立精确零分布;最后,从R'_k独立精确零分布中计算出R_k中每个候选对比序列模式的精确p-value值。

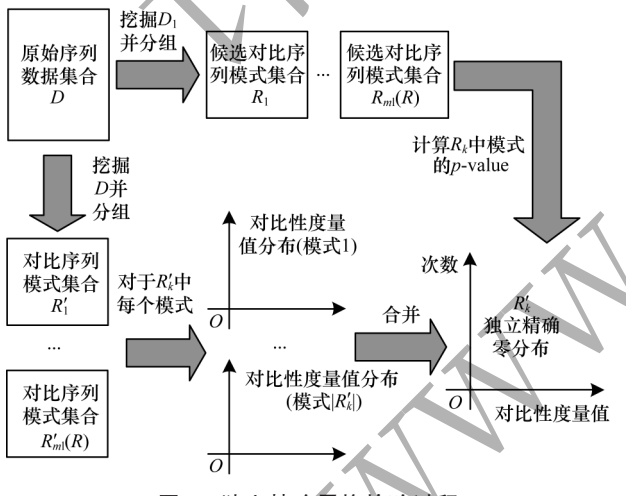


图2 独立精确置换检验过程

Fig.2 Process of independent exact permutation testing

独立精确置换检验最关键的步骤是每个模式对比性度量值分布的计算,该分布由对比性度量值和其在置换数据集合中的次数构成。给定一个t',t'的每个q'仅对应一个对比性度量值,即$\text{wra}(t',q')$。q'的最小值$L(t')$为$\min\{\theta_{\text{sup}},|D_1|+\sup(t',D_1)-|D|\}$,最大值$U(t')$为$\min\{\sup(t',D),|D_1|\}$,因此$q'\in[L(t'),U(t')]$。

t'的每个对比性度量值在置换数据集合中相应的次数,可以通过以下模拟置换过程计算得出:

$$g_1(t',q')=\binom{\sup(t',D)}{q'} \tag{2}$$

$$g_2(t',q')=\binom{|D|-\sup(t',D)}{|D_1|-q'} \tag{3}$$

$$n_1^{\text{npset}}(t',q')=g_1(t',q')g_2(t',q')(|D_1|)!(|D_2|)! \tag{4}$$

其中:$g_1(t',q')$表示从D中含有t'的序列中随机拿出q'条放入D'_1中;$g_2(t',q')$表示从D中不含t'的序列中随机拿出$|D_1|-q'$条放入D'_1中。因此,$g_1(t',q')$与$g_2(t',q')$相乘表示只有q'条序列含有t'的置换数据集合D'_1的数量。同时,再考虑D'_1和D'_2内部序列的排列可能性:$|D_1|!$ 和$|D_2|!$,式(4)的结果即是$\text{wra}(t',q')$值在置换数据集合中相应的次数。

对于R'_k中的每个模式t',运用式(1)和式(4)就能计算出t'对应的对比性度量值分布。因此,R'_k对应的独立精确零分布中对比性度量值的总数为:

$$t_{\text{total\_wra}}(R'_k)=\sum_{t\in R'_k}\sum_{q'=L(t')}^{U(t')}n_1^{\text{npset}}(t',q') \tag{5}$$

从而,R_k中每个候选对比序列模式t°的精确p-value值计算如下:

$$p_{p\text{-value}}(t^{\circ},k)=\frac{\sum_{t'\in R_k}\sum_{q'\in W}n_1^{\text{npset}}(t',q')}{t_{\text{total\_wra}}(R'_k)} \tag{6}$$

其中:W表示R'_k中比t°更极端的模式对应的序列数量分布集合,即$W=\{q'\mid \text{wra}(t^{\circ},\sup(t^{\circ},D_1)\leqslant\text{wra}(t',q'))\}$。

从式(6)可以得知,最终精确p-value值的计算公式的分子分母均为式(4)的累加结果。因此,为了减少计算开销,可以删去式(4)中的$|D_1|!$ 和$|D_2|!$项,即:

$$n_2^{\text{npset}}(t',q')=g_1(t',q')g_2(t',q') \tag{7}$$

3.3 约束度量

计算得到R_k中每个候选对比序列模式的精确p-value值后,IEP-DSP算法运用BH方法将R_k中的FDR度量值约束在置信度为α的统计显著水平下。具体而言,先将R_k中候选对比序列模式按照p-value值从小到大排序进行排列得到C_k,然后进行如下计算:

$$B_{\text{BH}}(C_k,\alpha)=\{c_i\mid p_{p\text{-value}}(c_i)\leqslant\frac{i\alpha}{|C_k|}\wedge c_i\in C_k\} \tag{8}$$

最终非统计显著的对比序列模式c_i将被过滤。

3.4 IEP-DSP算法步骤

根据以上讨论,详细的IEP-DSP算法步骤见算法1。

算法1 IEP-DSP($D,\theta_{\text{sup}},\theta_{\text{dis}},\alpha$)

输入 序列数据集合$D=\{D_1,D_2\}$;支持度阈值θ_{sup};对比性度量阈值θ_{dis},统计显著水平α

输出 统计显著的对比序列模式C^*

1.R← pattern_mining ($D_1,\theta_{\text{sup}},\theta_{\text{dis}}$)

2.R'← pattern_mining ($D,\theta_{\text{sup}},\theta_{\text{dis}}$)

3.$R_1,R_2,\cdots,R_{ml(R)} \leftarrow$ len_cla(R)
4.$R'_1,R'_2,\cdots,R'_{ml(R)} \leftarrow$ len_cla(R')
5.for k = 1 to ml(R) do
6.$I_k \leftarrow$ iend_generation(R'_k,θ_{sup})
7.end for
8.for k = 1 to ml(R) do
9.sort(I_k)
10.accumulate(I_k)
11.end for
12.for k = 1 to ml(R) do
13.for t^o in R_k do
14.x = find_wra(wra(t^o,sup(t^o,D_1)),I_k)
15.p-value(t^o,k) = x / last_nc(I_k)
16.end for
17.$C_k \leftarrow$ p_sort(redundancy_remove(R_k))
18.$C^*_k \leftarrow$ BH(C_k,α)
19.end for
20.$C^* \leftarrow$ union($C^*_1,C^*_2,\cdots,C^*_{ml(R)}$)
21.return C^*

算法1相应的解释如下:

1)运用pattern_mining()方法挖掘D_1中的候选对比序列模式并放入集合R(第1步);运用pattern_mining()方法挖掘D中的对比序列模式并放入集合R',R'中的模式即是所有可能在置换数据集合中出现的对比序列模式(第2步)。

2)运用len_cla()方法将R和R'中的模式根据长度进行分组(第3步、第4步)。对于每个R'_k,分别用iend_generation()方法建立其对应的独立精确零分布I_k(第5步、第7步)。

3)对于每个独立精确零分布I_k,根据z_{wr}值的降序排列所有<z_{wr},z_{nc}>对,并根据该顺序累加I_k中<z_{wr},z_{nc}>对的z_{nc}值(第8步~第11步)。上述操作是为了快速检索大于等于某个对比性度量值的WRAcc值的个数。每个I_k中最后一个<z_{wr},z_{nc}>对的z_{nc}值即是该独立精确零分布中所有的WRAcc值个数。

4)对于R_k中每个候选对比序列模式t^o,运用find_wra()方法找到比t^o更极端的模式数量x;随后t^o的精确p-value值可由x/last_nc(I_k)计算得出,其中last_nc()返回I_k中最后一个<z_{wr},z_{nc}>对的z_{nc}值(第12步~第16步)。

5)运用redundancy_remove()方法过滤R_k中冗余模式。这里的冗余模式指的是p-value值大于等于任一子模式的p-value值的候选对比序列模式;再运用p_sort()方法根据p-value值从小到大排序模式后,就能够使用BH()方法将每组R_k中的FDR控制在置信度为α的统计显著水平下,最终,合并所有C^*_k即得到统计显著的对比序列模式集合C^*(第17步~第20步)。

算法2的作用是为R'_k构建独立精确零分布。具体而言,计算R'_k中每个模式t'的所有q'对应的WRAcc值z_{wr}和其相应的数量z_{nc},并将每一对<z_{wr},z_{nc}>放入到集合I_k中。集合I_k的最终结果即是R'_k对应的独立精确零分布。

算法2 iend_generation(R'_k,θ_{sup})
输入 k长对比序列模式集合R'_k;支持度阈值θ_{sup}
输出 R'_k对应的置换检验独立精确零分布I_k
1.for t' in R'_k do
2.for q' = L(t') to U(t') do
3.$z_{wr} \leftarrow$ wra(t',q')
4.$z_{nc} \leftarrow$ $npset_2$(t',q')
5.$I_k = I_k \cup \{< z_{wr}, z_{nc} >\}$
6.end for
7.end for
8.return I_k

IEP-DSP算法各步骤的时间复杂度分析:频繁模式挖掘算法的时间复杂度分析见文献[23],其对IEP-DSP算法的时间复杂度影响不大;模式长度分组操作可以在模式数量的线性阶时间内完成;构建每个R'_k对应的独立精确零分布操作等同于计算R'中每个对比序列模式t'的对比性度量值分布,因此该操作的时间复杂度为$O(|R'|\text{avg}(U(t')-L(t'))$;排序操作和累加操作可以在统计度量值数量的线性对数阶和线性阶时间内完成;p-value值计算操作、去冗余操作和FDR计算操作均可在模式数量的线性阶时间内完成。从上述分析可知:IEP-CSP算法的时间复杂度主要由构建独立精确零分布操作决定,即$O(|R'|\text{avg}(U(t')-L(t'))$。

从式(1)和式(4)中可以发现,如果置换数据集合中2个对比序列模式t'_1和t'_2在D中的支持度相同,即sup(t'_1,D)等于sup(t'_2,D),那么t'_1和t'_2构建的对比性度量值分布就相同。为了减少IEP-DSP算法的时间复杂度,支持度相同的模式的对比性度量值分布只需计算1次即可。因此,IEP-DSP算法的时间复杂度减少为$O(|R'_{cs}|\text{avg}(U(t')-L(t'))$,其中$R'_{cs}$表示合并$R'$中所有支持度相同的对比序列模式的结果。

4 实验

为了验证IEP-DSP算法的有效性,在真实数据集和仿真数据集上进行了大量对比实验。对比的方法包括SP-DSP算法、DSPM-MTC算法[9]、ESM算法[7]和IMP算法[3]。其中,SP-DSP算法使用标准置换检验挖掘对比序列模式。在所有算法中,ESM算法和IMP算法是基于对比性度量约束的挖掘算法,IEP-DSP算法、SP-DSP算法和DSPM-MTC算法是基于统计显著性检验的挖掘算法,且这3个算法均使用FDR作为约束。所有实验均使用一台配置为2.40 GHz CPU和12 GB内存的电脑设备。

4.1 真实数据集实验

4.1.1 数据信息

实验选用了4个不同类型的真实数据集,即Epitope[24]、Unix[25]、Question[26]和Phospep[27]。Epitope是抗原蛋白序列的数据集;Unix是用户操作序列的

数据集；Question是文本序列的数据集；Phospep是磷酸化肽段序列的数据集。数据集的详细信息如表2所示，其中，k_{min}、k_{max}和k_{avg}分别表示序列最短长度、序列最长长度和序列平均长度。

表2 真实数据集信息

Table 2 Information of the real data sets

数据集	$\|E\|$	$\|D\|$	k_{min}	k_{max}	k_{avg}
Epitope	20	2 392	9	21	15.0
Unix	1 103	4 015	1	564	26.4
Question	3 612	1 731	4	29	10.2
Phospep	20	10 000	33	33	33.0

4.1.2 真实数据集实验结果

为评估每个算法的挖掘能力，本文首先对比了每个算法在相同参数下(θ_{sup},θ_{dis},α)报告的对比序列模式数量，结果如图3所示。从实验结果可以看出：基于统计显著性检验的方法得到的模式数量远小于基于对比性度量约束的方法，这是因为基于统计显著性检验的方法除了考虑对比性度量约束外，还会考虑统计显著性约束；在基于对比性度量约束的方法中，ESM算法得到的模式数量非常多，其原因是ESM算法没有使用去冗余的方法；在基于统计显著性检验的方法中，IEP-DSP算法比SP-DSP算法、DSPM-MTC算法报告的模式数量更多，这表明独立精确置换检验能够拒绝更多的零假设。

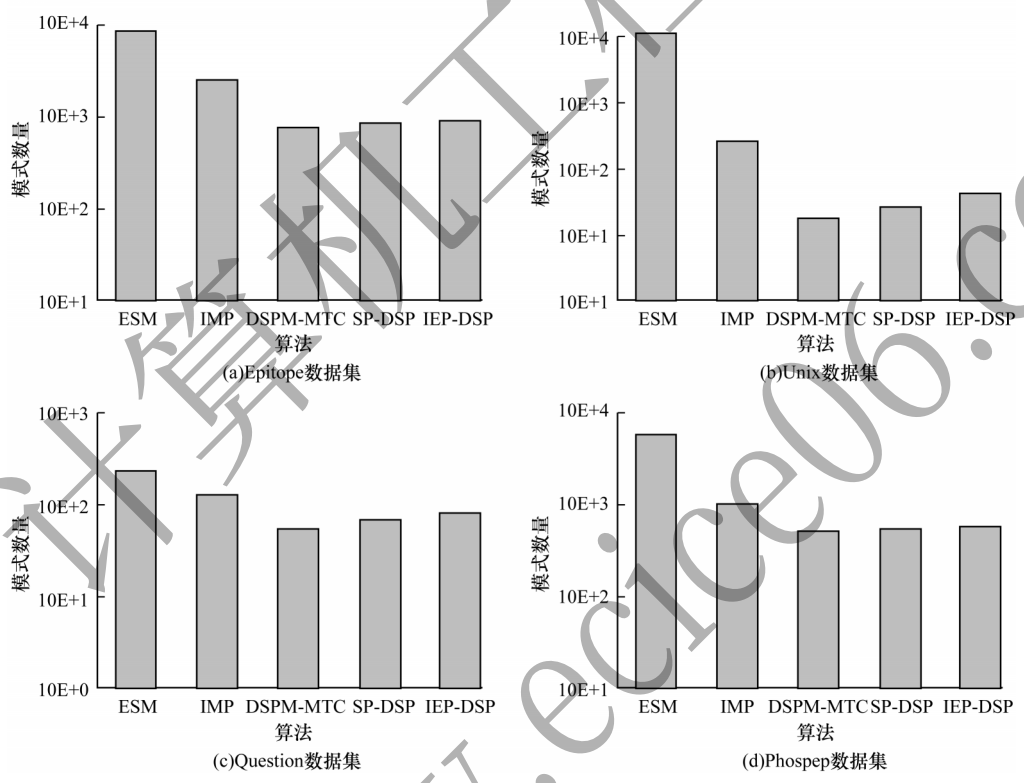


图3 每个算法在不同的数据集上报告的对比序列模式数量

Fig.3 Number of contrast sequential patterns reported by each method on different data sets

由于真实数据集中对比序列模式真假信息的缺失，不能直接根据各个算法报告的模式结果评价其有效性，因此后续实验采用一种间接的分类预测方法评估返回的模式质量[28]，即根据每个算法报告的模式信息，为数据集中的每条序列构建一个特征向量，将该向量送入分类器进行预测。特征向量的每个值是该序列和模式的包含关系，即包含为1，不包含则为0。该实验能够间接反映挖掘到的模式的真假性的原因是：真对比序列模式本质上体现了不同类型序列数据的相异性。为了减小分类器本身影响，实验采用了3种不同类型的分类器，分别为朴素贝叶斯、支持向量机和多层感知机。实验结果如表3~表5所示，每个正确率值均取自于10次预测结果的平均值。

表3 朴素贝叶斯分类器的分类正确率

Table 3 Classification accuracy reported by the Naive Bayes classifier

算法	Epitope	Unix	Question	Phospep
ESM	0.528	0.712	0.804	0.546
IMP	0.568	0.647	0.835	0.574
DSPM-MTC	0.614	0.912	0.865	0.597
SP-CSP	0.627	0.924	0.885	0.603
IEP-CSP	0.645	0.936	0.897	0.618

表4　支持向量机分类器的分类正确率

Table 4　Classification accuracy reported by the support vector machine classifier

算法	Epitope	Unix	Question	Phospep
ESM	0.581	0.725	0.828	0.583
IMP	0.721	0.882	0.846	0.591
DSPM-MTC	0.727	0.928	0.872	0.634
SP-DSP	0.734	0.939	0.887	0.643
IEP-DSP	0.756	0.949	0.901	0.665

表5　多层感知机分类器的分类正确率

Table 5　Classification accuracy reported by the multilayer perceptron classifier

算法	Epitope	Unix	Question	Phospep
ESM	0.616	0.748	0.844	0.571
IMP	0.748	0.898	0.862	0.584
DSPM-MTC	0.757	0.922	0.895	0.625
SP-DSP	0.762	0.927	0.913	0.638
IEP-DSP	0.786	0.940	0.924	0.653

从不同分类器的分类结果中可以看出:基于统计显著性检验的方法的分类正确率高于基于对比性度量约束的方法。因此,可以说明基于统计显著性检验的方法过滤了许多假阳性对比序列模式。以Question数据集为例,基于对比性度量约束的方法会返回<what,is>和<where,the>模式,而基于统计显著性检验的方法只有<what>和<where>模式。is和the在英文句子中出现频率很高,且通常作为语法结构出现,因此它们无法表现句子的差别,从而给分类器造成干扰。

基于统计显著性检验的3种算法的准确率高低排序为:IEP-DSP算法>SP-DSP算法>DSPM-MTC算法,这个结果证明了IEP-DSP算法能够保留更多的真对比序列模式。以Phospep数据集实验结果为例,IEP-DSP算法保留了<A,L,E,S>模式,而SP-DSP算法和DSPM-MTC算法只保留了<A,S>模式,从而导致7条包含<A,L,E,S>的磷酸化肽段被分类为非磷酸化肽段,此现象说明了<A,L,E,S>模式应该是真对比序列模式。综上,IEP-DSP算法不仅能够过滤大量假阳性模式,还能够尽可能地保留真对比序列模式。

4.1.3　IEP-DSP算法与SP-DSP算法

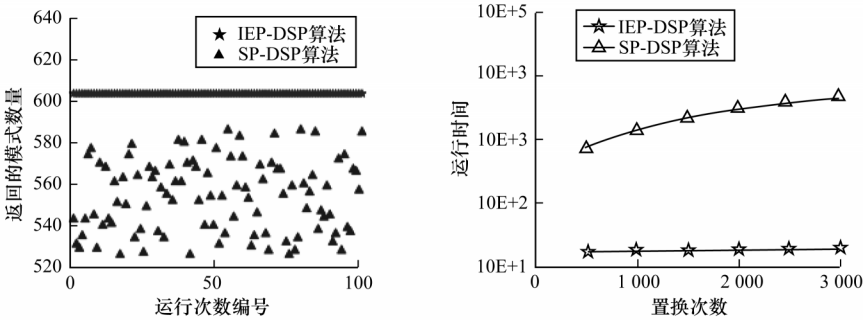
在2个置换检验算法中,IEP-DSP算法使用的是独立精确置换检验构建精确零分布,SP-DSP算法使用的是标准置换检验构建共享近似零分布。为了证明独立精确零分布能够去除共享近似零分布的4个缺点,本文进行了以下的讨论和实验。

在SP-DSP算法报告的结果中,存在一定数量p-value值为0的对比序列模式。这是因为SP-DSP算法生成的置换数据集合中没有找到比这些模式更为极端的模式存在。而在IEP-DSP算法报告的结果中,所有模式的p-value值均不为0。这是因为IEP-DSP算法考虑了所有的置换数据集合,总能找到至少和这些模式一样极端的模式存在。p-value值等于0是一个非常差的近似值,它表达的意义是这些模式的统计显著性无穷大。然而,在某些非常谨慎的应用中,即使α设置得非常小也无法过滤掉这些模式。

在SP-DSP算法中,不同长度模式的p-value值均通过同一个共享零分布计算得到;而在IEP-DSP算法中,不同长度模式的p-value值通过各自的独立零分布计算得到。在共享零分布中,子模式和超模式之间存在相应序列数据的反单调性,从而在计算p-value值时会存在一定程度的互相干扰,这个情况导致了SP-DSP算法报告的模式数量少于IEP-DSP算法。

图4(a)展示了在Phospep数据集上运行100次IEP-DSP算法和SP-DSP算法返回的结果。可以看出:SP-DSP算法结果会有波动,而IEP-DSP算法结果是唯一的。这是因为标准置换检验中置换数据集合的生成存在随机性,从而构建的近似零分布也存在随机性,而独立精确置换检验构建的每个独立精确零分布都是唯一的。标准置换检验的随机性导致了SP-DSP算法难以判定处于阈值边界的对比序列模式的统计显著性,可以采用多次运行取平均的方法,但这必然会导致计算开销的大幅提升。

图4(b)展示了在Unix数据集中IEP-DSP算法和SP-DSP算法的运行时间。可以看出:IEP-DSP算法的运行时间显著低于SP-DSP算法的运行时间,其原因是IEP-DSP算法不需要实际生成置换数据集合,而SP-DSP算法不仅需要实际生成一定次数的置换数据集合,还需要对置换数据集合进行挖掘。此外,对于不同的数据集合而言,很难确定需要执行多少次置换才能得到一个误差较小的近似零分布。为了得到更准确的近似零分布,SP-DSP算法需要增加置换次数,这会导致SP-DSP算法需要的更多的运行时间。



(a)在Phospep数据集上运行100次的结果 (a)在Unix数据集上的运行时间

图4 2种算法的对比序列模式数量和运行时间

Fig.4 Distinguishing seguential patterns number and running time of two algorithms

综上,IEP-DSP算法能够去除SP-DSP算法的4个缺点。这体现了独立精确置换检验相较于标准置换检验的优势。

4.2 仿真数据集实验

4.2.1 数据生成

由于真实数据集缺少对比序列模式真假的信息,实验生成了仿真实验数据进一步验证各个算法的有效性。仿真数据的生成步骤如下:

1)假设$E_{\text{false}}=\{e_1,e_2,\cdots,e_{30}\}$表示随机元素字母表,$E_{\text{true}}=\{e_{31},e_{32},\cdots,e_{42}\}$表示植入元素字母表。

2)从E_{false}中随机挑选元素生成4 000条长度为30的序列数据组成D_2;从D_2中随机挑选800条序列数据组成D_1。

3)从E_{true}中随机挑选6个字母作为长度为1的对比序列模式,并指定任意4个模式的支持度范围为150~190,余下2个模式支持度范围为40~80。6个模式的支持度的和等于800。为每个模式选择1个位置进行植入,具体做法是直接用模式替代D_1序列中相应元素,同时每条序列数据包含且只包含1个长度为1的植入模式。

4)从E_{true}中挑选未使用的4个字母同支持度最高的4个长度为1的模式结合生成长度为2的对比序列模式。其中,支持度最高的2个长度为1的模式生成的长度为2的模式支持度范围为110~150,其余2个生成的长度为2的模式支持度范围为40~80。植入方式同第3步。

5)从E_{true}中选择未使用的2个字母同支持度最高的2个长度为2的模式结合生成长度为3的对比序列模式,这2个长度为3的模式的支持度范围为40~80。植入方式同第3步。

通过上述步骤,人为植入了6个长度为1、4个长度为2和2个长度为3的对比序列模式。同时,在挖掘算法返回的对比序列模式中,如果某个对比序列模式包含E_{true}中的元素,则该模式被认定为真对比序列模式;反之,如果某个对比序列模式仅包含E_{false}中的元素,则该模式被认定为假阳性对比序列模式。

4.2.2 仿真数据实验结果

为减小随机性的影响,实验共生成了10组仿真数据集。各个算法返回的对比序列模式信息如表6所示,其中每个结果取自于10个仿真数据集挖掘结果的平均值。从表6可以看出,基于对比性度量约束的ESM算法和IMP算法都报告了许多对比序列模式,其中大部分模式为假阳性对比序列模式;而基于统计显著性检验的DSPM-MTC算法、SP-DSP算法和IEP-DSP算法报告的模式数量较少,且大部分为真对比序列模式。在这3种方法中,IEP-DSP算法报告的模式数量最多,且假阳性对比序列模式最少,这证明了IEP-DSP算法能过滤掉大量对比性度量约束方法中报告的假阳性模式,且相较于SP-DSP算法和DSPM-MTC算法能够保留更多的真对比序列模式,体现了IEP-DSP算法挖掘对比序列模式的优势。值得注意的是,ESM算法报告了许多真对比序列模式,这是因为ESM算法没有使用去冗余方法,从而导致了大量真对比序列模式实际上提供了重复的信息。

表6 不同算法的真对比序列模式和假阳性模式数量

Table 6 Number of true distinguishing seguential patterns and false positive patterns of different algorithms

算法	模式数量	真模式	假阳性模式
ESM	11 452.4	2 927.9	8 524.5
IMP	1 847.8	73.5	1 774.3
DSPM-MTC	55.1	52.5	2.6
SP-CSP	58.4	56.1	2.3
IEP-CSP	66.6	64.7	1.9

5 结束语

为过滤对比序列模式挖掘算法中存在的大量假阳性模式,本文提出一种面向对比序列模式的独立精确置换检验挖掘算法。该算法能为不同长度的模式分别构建独立精确零分布,从而能够计算出精确p-value值。实验结果表明,该算法不仅能够去除一定数量的假阳性对比序列模式,且能够比其他统计

显著性检验方法保留更多的真对比序列模式,验证了独立精确置换检验相较于标准置换检验的优越性。此外,本文算法倾向于保留较短的对比序列模式,主要是因为其采用了去冗余方法,即如果一个对比序列模式t的p-value值大于其任何一个子模式t_{sub}的p-value值,则该对比序列模式被认定为冗余模式。由于t和t_{sub}的支持度具备反单调性关系,因而t_{sub}会对t的统计显著性产生影响,但该影响不具备反单调性关系。单纯地运用p-value值比较方法能够去除掉一定数量的冗余模式,但是也会过滤掉一些非冗余模式。因此,下一步将研究更优的去除子对比序列模式统计显著性影响的方法。

参考文献

[1] 刘睿涛,陈左宁. 基于统计数据的超级计算机内存故障分析[J]. 计算机工程,2019,45(5):35-45.
LIU R T, CHEN Z N. Supercomputers memory faults analysis based on statistical data[J]. Computer Engineering, 2019,45(5):35-45. (in Chinese)

[2] 谢彬,张琨,蔡颖,等. 移动目标关联共现规则挖掘算法研究[J]. 计算机工程,2018,44(8):61-67,73.
XIE B, ZHANG K, CAI Y, et al. Research on mining algorithm for association co-occurrence rule of moving targets[J]. Computer Engineering, 2018, 44(8): 61-67, 73. (in Chinese)

[3] ZHENG Z G, WEI W, LIU C M, et al. An effective contrast sequential pattern mining approach to taxpayer behavior analysis[J]. World Wide Web, 2016, 19(4): 633-651.

[4] PANG T H, DUAN L, LI L, et al. Mining similarity-aware distinguishing sequential patterns from biomedical sequences [C]//Proceedings of the 4th International Conference on Data Science in Cyberspace. Shenzhen, China:[s. n.],2017:43-52.

[5] MICHELE D, BAIARDI F, LIPILINI J, et al. Sequential pattern mining for ICT risk assessment and management [J]. Journal of Logical and Algebraic Methods in Programming, 2019, 102(1): 1-16.

[6] 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,2018,5(2):680-686.
JIANG B, GU F Y, HE Z Y. Mining top-k non-redundant distinguishing sequential patterns[J]. CAAI Transactions on Intelligent Systems, 2018, 5(2): 680-686. (in Chinese)

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

[8] 王慧锋,段磊,左劼,等. 免预设间隔约束的对比序列模式高效挖掘[J]. 计算机学报,2016,39(10):1979-1991.
WANG H F, DUAN L, ZUO J, et al. Efficient mining of distinguishing sequential patterns without a predefined gap constraint[J]. Chinese Journal of Computers, 2016, 39(10):1979-1991. (in Chinese)

[9] HE Z Y, ZHANG S M, WU J. Significance-based discriminative sequential pattern mining[J]. Expert Systems with Applications, 2019, 122(1): 54-64.

[10] LIU G M, ZHANG H J, WONG L. Controlling false positives in association rule mining[J]. Proceedings of the VLDB Endowment, 2011, 5(2): 145-156.

[11] WU J, HE Z Y, GU F Y, et al. Computing exact permutation p-values for association rules[J]. Information Sciences, 2016, 346(1): 146-162.

[12] JUNPEI K, MASAKAZU I, HIROKI A, et al. Statistical emerging pattern mining with multiple testing correction [C]//Proceedings of the 23th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2017: 897-906.

[13] PELLEGRINA L, RIONDAT M, VANDIN F. Hypothesis testing and statistically sound pattern Mining [C]// Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2019: 3215-3216.

[14] BRIN S, MOTWANI R, SILVERSTEIN C. Beyond market baskets: generalizing association rules to correlations[C]// Proceedings of the 12th ACM SIGMOD International Conference on Management of Data. New York, USA: ACM Press, 1997: 265-276.

[15] ZHANG H, PADMANABHAN B, TUZHILIN A. On the discovery of significant statistical quantitative rules[C]// Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2004: 374-383.

[16] WEBB G I. Layered critical values: a powerful direct-adjustment approach to discovering significant patterns[J]. Machine Learning, 2008, 71(2/3): 307-323.

[17] TERADA A, KIM H, SESE J. High-speed westfall-young permutation procedure for genome-wide association studies [C]//Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics. New York, USA: ACM Press, 2016: 17-26.

[18] LEONARDO P, FABIO V. Efficient mining of the most significant patterns with permutation testing [C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2018: 2070-2079.

[19] PELLEGRINA L, RIONDAT M, VANDIN F. SPUMANTE: Significant pattern mining with unconditional testing[C]// Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2019: 1528-1538.

[20] FOURNIER P, LIN J, KIRAN R, et al. A survey of sequential pattern mining[J]. Data Science and Pattern Recognition, 2017, 1(1): 54-77.

[21] CARMONA C J, JESUS M J, HERRERA F. A unifying analysis for the supervised descriptive rule discovery via the weighted relative accuracy [J]. Knowledge-Based Systems, 2018, 139(2): 89-100.

[22] DAVID R, ABBAS R. Correcting false discovery rates for their bias toward false positives [J]. Communications in Statistics-Simulation and Computation, 2019, 12(1): 1-15.

(下转第61页)

（上接第53页）

[23] ZAKI M J. SPADE: an efficient algorithm for mining frequent sequences[J]. Machine Learning, 2001, 42(1/2): 31-60.

[24] DENG K, ZAÏANE O R. An occurrence based approach to mine emerging sequences [C]//Proceedings of the 12th International Conference on Data Warehousing and Knowledge Discovery. Berlin, Germany: Springer, 2010: 275-284.

[25] DUA D, GRAFF C. UCI machine learning repository [EB/OL]. [2020-05-05]. http://archive. ics. uci. edu/ml.

[26] KIM Y. Convolutional neural networks for sentence classification [C]//Proceedings of the 18th Conference on Empirical Methods in Natural Language Processing. Doha, Qatar: ACL Press, 2015: 1745-1751.

[27] UNIPROT CONSORTIUM. The universal protein resource [J]. Nucleic Acids Research, 2007, 35(1): 193-197.

[28] ZHOU C, CULE B, GOETHALS B. Pattern based sequence classification[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 28(5): 1285- 1298.

编辑　索书志