带弱通配符的模式匹配及其在时序分析中的应用

檀朝东[1] 闵 帆[2] 吴 霄[1] 李欣伦[1]
(中国石油大学(北京)石油工程学院 北京 102249)[1] (西南石油大学计算机科学学院 成都 610500)[2]

摘 要 针对模式匹配的准确性和灵活性问题,提出了一种基于弱通配符的匹配算法,以快速定位重要的时间点,辅助用户决策。首先通过数据预处理得到编码字符串序列,然后定义具有特殊语义的弱通配符及区间长度,最后设计一种高效的模式匹配算法。在时序分析中,模式反映了数据的变化趋势,预示着事件的发生。传统的精确匹配受噪声的影响比较大,匹配的灵活性低。通过添加弱通配符可以兼顾匹配过程的灵活性和准确性。油田产量与股票交易数据实验表明,所提方法较精确匹配而言,能够更有效地找到符合用户要求的模式。

关键词 模式匹配,时间序列,弱通配符,数据预处理
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.01.016

Pattern Matching with Weak-wildcard in Application of Time Series Analysis

TAN Chao-dong[1] MIN Fan[2] WU Xiao[1] LI Xin-lun[1]
(College of Petroleum Engineering, China University of Petroleum-Beijing, Beijing 102249, China)[1]
(College of Computer Science, Southwest Petroleum University, Chengdu 610500, China)[2]

Abstract This paper proposed a pattern matching method based on weak-wildcards to obtain accurate and flexible matching which is good for locating critical time points and assisting users' decision. First, a nominal sequence was obtained through coding the time series. Second, the concepts of weak-wildcard and gaps with special semantics were defined. Third, an efficient pattern matching algorithm was designed. In time series analysis, patterns reflect the trend of data change and indicate the occurrence of events. The traditional exact pattern matching is greatly affected by the noise, which has lower matching flexibility. Adding weak-wildcards gives consideration to both accuracy and flexibility. Experiments were undertaken on oil production and stock transaction data. Results show that compared to exact pattern matching method, the proposed pattern matching method copes with users' expectation better.

Keywords Pattern matching, Time series, Weak-wildcard, Data preprocessing

1 引言

在现实世界中时间序列(简称时序)广泛存在,如油井的生产参数[1]、股票的交易记录[2]、气温的观测记录、电网的载荷、生物分子序列[3]等。时序所涉及的领域也十分广泛,包括气象、生物、石油化工等自然科学,资源勘探、地震预测等工程科学,金融分析[4]、人口统计等社会经济领域。时序蕴含着被观测事物与时间相关的重要信息,反映了一系列相邻观测值之间的关联性。常见的时序建模与分析方法包括线性回归、曲线拟合和参数估计[5]。时序分析正广泛应用于油田产量预测、股价分析、气象预报、电网预警等。

在时序分析中,模式反映了数据的变化趋势,预示着事件的发生。模式匹配能够快速定位重要的时间点,辅助用户决策。经典的精确匹配要求模式与子序列完全相同,缺乏灵活性[6]。在实际情况中,噪声会影响时序数据精确匹配的成功率。为此,研究者引入了通配符及区间的概念,使模式的形式更灵活,符合实际应用的需要;通配符的长度和区间大小则控制模式的灵活程度。

近年来,带通配符的模式匹配在国内外得到了广泛关注,并取得了一定研究成果。Fischer 等[7]最早提出了带固定数量通配符的模式匹配算法。随后,降低模式匹配时间复杂度的算法被提出[8-9]。为了放宽对模式中通配符的约束,Manber 等[10]研究了带可变长度通配符的模式匹配,但是他们研究的方法只能处理单个可变长度的通配符。Wu 等[11]提出了一种基于可变通配区间的网络树匹配算法。Min 等[12]针对指定间隔通配符的模式匹配,提出了一种高效的算法来计算匹配数量。Chen 等[13]提出了一种快速的算法,通过对连续模式的约束和匹配字符串长度的限制,使用户的查询具有广泛的灵活性。

本文提出一种基于弱通配符[14]的模式匹配算法,以解决

到稿日期:2017-03-03 返修日期:2017-06-08 本文受国家自然科学基金(61379089)资助。
檀朝东(1968-),男,博士,副研究员,主要研究方向为油田大数据挖掘,E-mail:tanchaodong@cup.edu.cn(通信作者);闵 帆(1973-),男,博士,教授,CCF 会员,主要研究方向为机器学习、数据挖掘,E-mail:minfanphd@163.com;吴 霄(1990-),男,硕士生,主要研究方向为油田大数据挖掘,E-mail:819565848@qq.com;李欣伦(1993-),男,硕士生,主要研究方向为油田大数据挖掘,E-mail:632977047@qq.com。

匹配的准确性和灵活性问题。首先，通过预处理将时序的数值数据转换为编码字符串序列，这种转换是基于数据的线性变化的，将所有数值的波动转化为不同的字母，得到时序的字符串序列。然后，定义了在模式匹配中具有特殊语义的新模式，一种是在字符串间添加通配符，另一种是在字符串间添加对通配符加以限制的弱通配符。通配符允许模式中存在噪声，比一般的模式更加灵活。弱通配符使得模式在语义上更加丰富，它不仅允许模式中存在噪声，还能够对噪声的程度进行控制，其意义在于，不会将剧烈变化错误地当成噪声。最后，本文提出了一种基于弱通配符的模式匹配算法。

本文实验使用了两个数据集，包括 20 口油井生产一年以上的产量数据和 10 支股票两年的交易数据。实验结果表明，所提算法是有效的，带有弱通配符的模式匹配可以准确地找到匹配位置，更能满足在实际应用中的需要，有助于从不同视角来表征和描述石油数据和股票数据，并且可以有效地分析数据规律。

2 相关概念

为了更好地引出问题以及算法，本节给出了在模式匹配中用到的一些基本定义。

定义 1 字母表 Σ 是一个由字母组成的集合。

例如在文本的研究中，英文的字母表既包含了基础字母，又包含了各种符号；中文的字母表是单个汉字或符号。

定义 2 序列 $S=s_1s_2\cdots s_n$ 是指一系列有序排列的字母。

例如，序列 $S=bbccbccdcda$，它的字母表 $\Sigma=\{a,b,c,d\}$。

更广泛地，序列既可以基于字母表，也可以基于整数、实数集合等。其中包含了一种特殊的序列，它与时间相关，被称作时间序列。

定义 3 时序 $T=\{t_i\mid i=1,2,\cdots,n\}$ 是一串按时间先后顺序有序排列的观测值，其中 n 是观测值的个数，t_i 是在第 i 个时刻的观测值。

例如，油井生产时序 $T=\{10.1,9.8,10.1,9.9,9.8\}$，这是某油田中的一口井 5 天的产量，它形成了一个长度为 5 的时序。时序的一个基本特征是相邻的观测值之间存在相互依赖性，数据有序性则是这种依赖性的一种表现形式。

实验中收集整理了 20 口油井的数据和 2 年的股票数据，通过去掉其中缺失的部分时间段的数据，得到了相应字母组成的时序。

由于本文只对名词型数据进行匹配，对于其他类型的数据，需要通过预处理将其转化为名词型数据。对于名词型的时序，可以将每一个时刻的观测值看作一个字符，则一个时序就是由许多字符按顺序连接起来的序列。任意序列 $S=s_1s_2\cdots s_n$，s_i 表示第 i 个时刻的字符，序列 S 的长度即为该序列包含的字符的个数，记作 $|S|$，例如一个序列 $S=bbccbccdcda$，那么 $|S|=11$。

在分析一个序列时，通常更关注该序列的一个小部分，即所谓的模式。例如单词可以看作一篇文章的模式，每一个字母又可以看作单词的模式。通常来说，一个模式 $P=p_1p_2\cdots p_m$ 也可以当作一个序列。

定义 4 模式 $P=p_1p_2\cdots p_m$ 是字母组成的串。

定义 5 模式匹配。给定序列 $S=s_1s_2\cdots s_n$ 和模式 $P=p_1p_2\cdots p_m$，若在某一个位置 i，对于 $\forall 1\leqslant j\leqslant m$，都有 $s_{i+j-1}=p_j$，则称模式 P 在 S 的位置 i 处与 S 匹配。例如：给定一个序列 $S=bbccbccdcda$，模式 $P=bc$，则模式 P 在 S 中匹配 1 次。

定义 6 通配符是一个可以匹配序列中的任意字符的特殊字符，记作 ϕ。使用 $[N,M]$ 表示通配符的可变长度，即可以插入的任意匹配字符的个数，其中 N 和 M 分别是通配符的最小长度和最大长度。一个带有通配符的模式可以写成 $P=p_1[N,M]p_2[N,M]\cdots[N,M]p_m$，其中 m 是模式的长度。

例如：对于给定的一个序列 $S=bbccbccdcda$，模式 $P=b[1,3]c$，则该模式在原序列中共匹配了 6 次，位置为：$\langle1,3\rangle$，$\langle1,4\rangle$，$\langle2,4\rangle$，$\langle2,6\rangle$，$\langle5,7\rangle$ 和 $\langle5,9\rangle$。加入通配符的概念可以使得模式匹配具有一定的灵活性，从而更加满足在实际情况中的应用。

定义 7[14] 设 Ω 为一些特定的弱的字符的集合，弱通配符 ψ 表示它可以匹配 Ω 中的任意字符，但并不能匹配整个序列中的所有字符。弱通配符是序列中的一些变化小或者影响不大的字符。采用区间 $[N,M]$ 表示弱通配符的可变长度，N 是弱通配符的最小长度，M 是弱通配符的最大长度。简便起见，假设弱通配符的最小长度 N、弱通配符的最大长度 M 和弱的字符的集合 Ω 是不变的。

3 带通配符的模式匹配

本节首先讨论数据转换，将原始数据转换为能用于模式匹配的名词型序列，然后提出本文的问题，之后展示求得成功匹配次数以及匹配位置的两个算法的伪代码。

3.1 数据预处理

数据预处理是指在数据挖掘前对其进行的一些处理，包括数据归一化、特征标准化等方法，是整个数据挖掘过程中的一个重要步骤。目前，数据预测方法一般是基于数据拟合的方法，其算法存在一定的局限性，为了更好地掌握一口井或者股价的特点，支持专家对数据状态的分析。根据行业特点，数据的波动比数据本身更具分析价值，因此使用了一个将数值型时序转化为名词型序列的编码表，来体现前后数据的变化。

给定一个时间序列 $T=\{t_i\mid i=1,2,\cdots,n+1\}$，则从时序 i 到时序 $i+1$ 的波动为 $f_i=(t_{i+1}-t_i)/t_i$，其中 $1\leqslant i\leqslant n+1$。为了能够给时间序列编码，提出了线性编码表，即间隔是线性增加的，如表 1 所列。根据线性编码表得到的字母表 $\Sigma=\{A,B,C,D,E,F,O,a,b,c,d,e,f\}$，将时间序列 $T=\{t_i\mid i=1,2,\cdots,n+1\}$ 转化为序列 $S=s_1s_2\cdots s_n$。

表 1 线性编码表

Table 1 Linear coding table

f_i	编码	f_i	编码
(−1%,1%)	O	(−3%,−1%)	a
(1%,3%)	A	(−7%,−3%)	b
(3%,7%)	B	(−13%,−7%)	c
(7%,13%)	C	(−21%,−13%)	d
(13%,21%)	D	(−31%,−21%)	e
(21%,31%)	E	(−100%,−31%)	f
(31%,inf)	F		

3.2 弱通配符模式

带弱通配符的模式是一个由字符和弱通配符区间组成的

序列，即 $P=p_1[N,M]p_2[N,M]\cdots[N,M]p_m$，其中 m 是模式的长度。

例如对于给定的一个序列 $S=bbccbccdcda$，$\Omega=\{a,b\}$，带弱通配符的模式 $P=b[0,2]c[0,2]c$，该模式在 S 中一共匹配了 3 次，位置分别为〈1,3,4〉，〈2,3,4〉，〈5,6,7〉。若给出的是带通配符的模式 $P=b[0,2]c[0,2]c$，则会多进行 2 次匹配，分别位于〈2,4,6〉和〈5,7,9〉。

3.3　问题定义与分析

本小节提出了一个新的模式问题。

问题 1　带弱通配符的模式匹配

输入：时序 T，时间点 t（一般是当前时间，从该时间提取模式），通配符的长度界限 N 和 M，弱通配符字母表 Ω，模式长度 $|P|$。如果直接给定模式，就只需要时序和模式。

输出：成功匹配的起始点集合 I，总的匹配次数 sum。

根据弱通配符的定义可以看出，使用弱通配符可以准确地表示出在数据产生过程中噪声的影响程度，以便用户进行定位及辅助决策。弱通配符的模式匹配问题具有以下两点性质。

性质 1　从序列中任何一点开始的带弱通配符的模式 P 的匹配次数最多只有一次。

证明：设模式 P 的第一个字符 p_1 与序列 S 上的第 i 个位置 s_i 发生匹配，序列的下一个位置 s_{i+1} 若不是弱字符，则要求该位置与模式 P 的 p_2 匹配，否则第 i 个位置将匹配失败；若该位置是弱字符，则验证下一个位置 s_{i+2}。同样地，位置也可以分两种情况考虑，只有当模式 P 的所有位置都完成匹配时，模式 P 在序列的第 i 个位置才匹配成功，因此从序列的任何一点开始的带弱通配符的模式 P 的匹配次数最多只有一次。

与带弱通配符的模式匹配类似，模式的精确匹配有性质 1，但是带通配符的匹配不具有该性质。

性质 2　给定序列 $S=s_1s_2\cdots s_n$，模式 $P=p_1p_2\cdots p_m$，设精确匹配的次数为 AP，带弱通配符的模式匹配次数为 GP，带通配符的模式匹配次数为 WP，则有 $AP<=GP<=WP$。

证明：序列上的任何一个字符 s_i 在匹配模式时，若与模式 P 精确匹配，则在该位置带弱通配符和通配符的模式匹配也是成功的；若 $s_i\in\Omega$，则精确匹配失败，而带弱通配符和通配符的匹配仍是成功的；若 $s_i\notin\Omega\cap s_i\notin\emptyset$，那么只有带通配符的模式匹配是成功的。因此有 $AP<=GP<=WP$。

3.4　算法设计

针对提出的问题，设计了带弱通配符的模式匹配算法。算法 1 为程序的主程序，用于计算模式匹配的起始点位置和匹配次数。算法 2 为主程序调用的子程序，实现了添加弱通配度的模式匹配过程。

算法 1 通过给定一个序列 S，模式 P，求出成功匹配的序列起始点位置集合 I 和成功匹配的次数 sum。

算法 1　带弱通配符的模式匹配 WeakWildcardPatternMatch( )

```
输入：序列 S=s1s2…sn 和模式 P=p1p2… pm
输出：成功匹配的起始点集合 I，成功匹配的次数 sum
1. I=∅;/* 初始化集合 I */
2. sum=0;/* 初始化变量 sum */
3. for (i ← 1 to n−m+1) do
4. count←tailMatch(i,P,1);
   /* 遍历整个序列，调用 tailMatch 方法 */
5. if count > 0 then
6. I=I ∪ {i};/* tailMatch 方法的返回值大于 0，说明发生成功匹配，
   将序列的起始位置存放在集合 I */
7. end if
8. sum ← sum+count;/* sum 累计匹配次数 */
9. end for
10. return I,sum;
```

算法 1 的时间复杂度为 $O(n\times L\times m)$，n 为序列的长度，L 为通配符区间长度，m 为通配符长度，$m=M-N+1$。

算法 2 为算法 1 所调用的子程序，其功能是在模式的首字符匹配时递归调用本方法，返回这个位置成功匹配的次数。方法的参数有序列上的起始点 $start$、模式 P 以及在模式上的偏移量 $offSet$。

算法 2　模式的尾串匹配 tailMatch(start,P,offSet)

```
输入：序列的起点 start，弱通配符的最小程度 l 和最大长度 u，弱字符
      集合 Ω
输出：尾串成功匹配的次数 count
1. count=0;//初始化 count */
2. if offSet=|P| then
3. return 1;/* 若偏移量等于模式的长度，则说明匹配成功，返回值
   1 */
4. end if
5. for (i ← l+1 to u+1)do
6. if start+i > |S| then
7. break;/* 超出长度，跳出循环 */
8. end if
9. if P_offSet=S_start+i then
10. count ← count+tailMatch(start+i,P,offset+1);
    /* 序列上的当前位置与模式发生匹配，则递归调用 tailMatch 方
    法，判断后续位置是否匹配 */
11. end if
12. if P_offSet∉ Ω then
13. break;/* 序列上的当前位置的字符不在 Ω 中 */
14. end if
15. end for
16. return count;/* 返回成功匹配的次数 count */
```

4　实验结果

4.1　数据集

在实验过程中主要使用了 20 口油井一年的产量数据和股票的两年股价数据。

数据集 1：根据 20 口油井一年的产量数据进行了实验，每口井有约 350 个数据点。结果展示的数据集有 340 个数据点，最大值为 7.91，最小值为 2.49，平均值为 3.82。由于实际生产中的大部分时间比较平稳，因此字符 O 出现了 45 次，字符 A 出现了 29 次，字符 B 出现了 42 次，字符 a 出现了 32 次，字符 b 出现了 53 次，产量变化在 7% 以内的数据占数据集的 59%。

数据集 2:根据 10 支股票两年的收盘数据,对数据进行编码和模式匹配分析。结果展示的数据集有 635 个数据点,最大值为 17.98,最小值为 7.67,平均值为 11.57。根据股市的规则,只会出现 a,b,c,O,A,B,C 这 7 个字符,其中字符 O 出现了 112 次,字符 A 出现了 73 次,字符 a 出现了 80 次,股价变化在 3%以内的数据占数据集的 42%。

4.2 实验结果

对同一口井的一年的产量数据选取不同模式长度和不同通配符的匹配结果,如表 2—表 4 所列。实验中选择字母表 $\Sigma_1=\{a,b,c,d,e,f,O,A,B,C,D,E,F\}$,添加弱通配符的字符集合 $\Omega_1=\{a,O,A\}$,通配符 ϕ 的长度为(0,2),弱通配符 Ψ 的长度为(0,2)。

表 2 添加长度为(0,2)的通配符的匹配次数

Table 2 The number of matches with wildcard of increased length (0,2)

模式	匹配次数	运行次数
$P_3=a\phi O\phi A$	40	1531
$P_4=a\phi O\phi A\phi B$	14	2073
$P_5=b\phi a\phi O\phi A\phi B$	11	2465
$P_6=b\phi a\phi O\phi A\phi B\phi C$	6	2821

表 3 添加长度为(0,2)的弱通配符的匹配次数

Table 3 The number of matches with weak wildcard of increased length (0,2)

模式	Ω	匹配次数	运行次数
$P_3=a\Psi O\Psi A$	Ω_1	24	1336
$P_4=a\Psi O\Psi A\Psi B$	Ω_1	8	1767
$P_5=b\Psi a\Psi O\Psi A\Psi B$	Ω_1	5	2183
$P_6=b\Psi a\Psi O\Psi A\Psi B\Psi C$	Ω_1	2	2585

表 4 不添加通配符的匹配次数

Table 4 The number of matches without wildcard

模式	匹配次数	运行次数
$P_3=aOA$	8	1215
$P_4=aOAB$	3	1395
$P_5=baOAB$	0	1560
$P_6=baOABC$	0	1570

对比表 2—表 4 可以发现,不添加通配符的精确匹配能够匹配到的模式非常少,这种匹配方法非常不灵活,在长度逐渐增加后难以匹配到模式;而添加通配符的匹配则能匹配到较多的模式,通过弱通配符可以逐渐进一步增加匹配的精确性,并且不缺乏灵活性。

选取时间点为 7,模式长度为 5,从而得到模式 $OBBOc$,该模式表示油井产量有一定上涨后再下降的趋势。对 100 天的产量数据添加长度为(0,2)的通配符、长度为(0,2)且 $\Omega=\{a,O,A\}$的弱通配符和不添加通配符的匹配位置,从而得到匹配效果图,如图 1—图 3 所示。

对比图 1—图 3 可以发现,添加长度为(0,2)的通配符可以匹配到诸如第 17 天、第 66 天这种中间有一定的延迟且变化较大的时间,但也包含了模式 $OAAOb$ 所表达的趋势。而添加弱通配符的匹配可以通过定义 Ω,将含有间隔变化大的模式过滤掉,匹配到比较平稳的模式。若不添加通配符,则可以匹配到符合要求但数量非常有限的模式。总的来说,通过添加弱通配符可以进行更加有效的匹配,准确地找到匹配数量以及匹配位置。

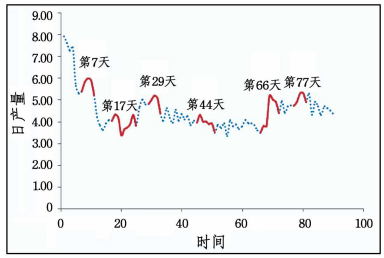

图 1 添加长度为(0,2)的通配符的匹配位置

Fig. 1 Matching positions with wildcard of increased length (0,2)

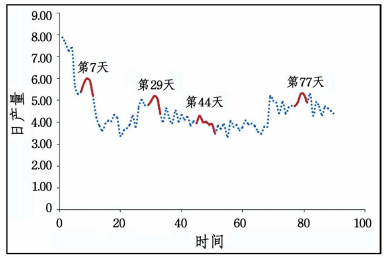

图 2 添加长度为(0,2)的弱通配符的匹配位置

Fig. 2 Matching positions with weak wildcard of increased length (0,2)

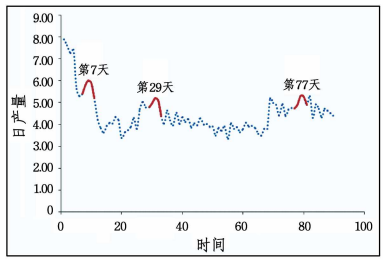

图 3 不添加通配符的匹配位置

Fig. 3 Matching positions without wildcard

针对同一支股票两年的收盘数据,选取不同的模式长度和不同的通配符,匹配结果如表 5—表 7 所列。实验中选择的字母表 $\Sigma_1=\{a,b,c,O,A,B,C\}$,添加弱通配符的字符集合 $\Omega_2=\{a,O,A\}$,通配符 ϕ 的长度为(0,2),弱通配符 Ψ 的长度为(0,2)。

表 5 添加长度为(0,2)的通配符的匹配次数

Table 5 The number of matches with wildcard of increased length (0,2)

模式	匹配次数	运行次数
$P_3=a\phi O\phi A$	35	2037
$P_4=a\phi O\phi A\phi A$	14	2359
$P_5=b\phi a\phi O\phi A\phi A$	8	2581
$P_6=b\phi a\phi O\phi A\phi B\phi A$	6	2773

表 6 添加长度为(0,2)的弱通配符的匹配次数

Table 6 The number of matches with weak wildcard of increased length (0,2)

模式	Ω	匹配次数	运行次数
$P_3=a\Psi O\Psi A$	Ω_1	23	1558
$P_4=a\Psi O\Psi A\Psi B$	Ω_2	10	1860
$P_5=b\Psi a\Psi O\Psi A\Psi B$	Ω_2	7	2176
$P_6=b\Psi a\Psi O\Psi A\Psi B\Psi C$	Ω_2	3	2639

表 7　不添加通配符的匹配次数

Table 7　The number of matches without wildcard

模式	匹配次数	运行次数
$P_3=aOA$	3	761
$P_4=aOAB$	1	867
$P_5=baOAB$	0	930
$P_6=baOABB$	0	1041

数据集 2 反映的规律与数据集 1 类似，并且对于股票交易数据而言，更小的数据波动意味着更少的精确匹配。

结束语　本文提出了一种适用于模式匹配的新方法。具体来说，先设计一个将时间序列转换为序列的编码表，然后定义弱通配符以及模式匹配的算法。对油井和股票数据的实验结果表明，这种新算法所匹配到的模式是符合要求的，匹配位置是准确的，匹配次数满足 $AP \leqslant GP \leqslant WP$，因此该方法对时序的相似性分析以及后续的数据预测是非常有意义的。

从所提算法以及进一步的研究来看，如何对匹配到的模式进行挖掘和分析是该算法的重要意义所在。由于大部分事物都具有多个属性，即含有多个时间序列，若能选取多个时间序列进行模式匹配，就可以匹配到更加精确的模式，更有利于开展后续工作。在这项工作中如何选择时间节点，以及如何提取专家模式来进行更合适的匹配是未来需要解决的问题。

参 考 文 献

[1] MONTGOMERY D C, JENNINGS C L, KULAHCI M. Introduction to time series analysis and forecasting[M]. John Wiley & Sons, 2015.

[2] SAKURAI Y, FALOUTSOS C, YAMAMURO M. Stream monitoring under the time warping distance[C]// 2007 IEEE 23rd International Conference on Data Engineering. IEEE, 2007: 1046-1055.

[3] HE Y, WU X, ZHU X, et al. Mining frequent patterns with wildcards from biological sequences[C]// 2007 IEEE International Conference on Information Reuse and Integration. IEEE, 2007: 329-334.

[4] LU C J, LEE T S, CHIU C C. Financial time series forecasting using independent component analysis and support vector regression[J]. Decision Support Systems, 2009, 47(2): 115-125.

[5] KEOGH E, KASETTY S. On the need for time series data mining benchmarks: a survey and empirical demonstration[J]. Data Mining and Knowledge Discovery, 2003, 7(4): 349-371.

[6] YANG Q, WANG X. 10 Challenging Problems in data mining research[J]. International Journal of Information Technology and Decision Making, 2006, 5(4): 597-604.

[7] FISCHER M J, PATERSON M S. String-matching and other products[C]// Proceeding of the 7th SIAM AMS Complexity of Com-putation. Cambridge, USA, 1974: 113-125.

[8] INDYK P. Faster algorithms for string matching problems: matching the convolution bound[C]// 39th Annual Symposium on Foundations of Computer Science. IEEE, 1998: 166-173.

[9] KALAI A. Efficient pattern-matching with don't cares[C]// Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms. Philadelphia, PA, USA: ACM, 2002: 655-656.

[10] MANBER U, BAEZA-YATES R. An algorithm for string matching with a sequence of don't cares[J]. Information Processing Letters, 1991, 37(3): 133-136.

[11] WU Y, WU X, MIN F, et al. A Nettree for pattern Matching with flexible wildcard Constraints[C]// IEEE International Conference on Information Reuse and Integration. IEEE, 2010: 109-114.

[12] MIN F, WU X, LU Z. Pattern Matching with Independent Wildcard Gaps[C]// Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing. 2009: 194-199.

[13] CHEN G, WU X, ZHU X, et al. Efficient string matching with wildcards and length constraints[J]. Knowledge & Information Systems, 2006, 10(4): 399-419.

[14] TAN C D, FAN M, WANG M, et al. Discovering patterns with weak-wildcard gaps[J]. IEEE Access, 2016, 4: 4922-4932.

（上接第 102 页）

[16] LI J, LI D, ZHANG Y. Efficient Distributed Data Clustering on Spark[C]// IEEE International Conference on CLUSTER Computing. Chicago: IEEE Computer Society, 2015: 504-505.

[17] SINHA A, JANA P K. A novel K-means based clustering algorithm for big data[C]// International Conference on Advances in Computing, Communications and Informatics. Jaipur: IEEE, 2016: 1875-1879.

[18] JIN C, LIU R, HENDRIX W, et al. A Scalable Hierarchical Clustering Algorithm Using Spark[C]// IEEE First International Conference on Big Data Computing Service and Applications. San Francisco Bay: IEEE, 2015: 418-426.

[19] SARAZIN T, AZZAG H, LEBBAH M. SOM Clustering Using Spark-MapReduce[C]// IEEE International Parallel & Distributed Processing Symposium Workshops. Phoenix: IEEE Computer Society, 2014: 1727-1734.

[20] CONRAD J G, AL-KOFAHI K, ZHAO Y, et al. Effective document clustering for large heterogeneous law firm collections[C]// Proceedings of the 10th international conference on Artificial intelligence and law. Bologna: ACM, 2005: 177-187.

[21] UCI Machine Learning Repository [DB/OL]. http://archive.ics.uci.edu/ml.

[22] Wikipedia Weka (machine learning) [CP/OL]. http://en.wikipedia.org/wiki/Weka, 2010.

[23] xj1986. MR-DBSCAN [EB/OL]. [2013-5-15]. https://github.com/xj1986/MR-DBSCAN.