Hans Journal of Data Mining 数据挖掘, 2016, 6(4), 179-191
Published Online October 2016 in Hans. http://www.hanspub.org/journal/hjdm
http://dx.doi.org/10.12677/hjdm.2016.64020

Hans汉斯

A Novel Timing Calculation Algorithm Based on Statistical Extremum for the Time Series of Process Object

Tonglin Zhu[1], Tao Du[1], Shouning Qu[1,2], Lianjiang Zhu[2]

[1]School of Information Science and Engineering, University of Jinan, Jinan Shandong
[2]Information Network Center, University of Jinan, Jinan Shandong
Email: eo_dut@ujn.edu.cn

Received: Oct. 9th, 2016; accepted: Oct. 24th, 2016; published: Oct. 27th, 2016

Abstract

In this paper, an algorithm for computing timing relationship among each link of the process object is proposed, and the validity of the algorithm is proved through the theoretical analysis. The algorithm is designed based on statistical time distance among extremum points of sampling data set of the process industry, can calculate the delay time between any two time series, and then get timing relationship between any two links. At the same time, experiments with sampling data set of the process industry demonstrates that the algorithm can obtain the delay time interval among time series and the timing relationship between each link of process object.

Keywords

Process Object, Data Mining, Time Series, Statistics, Extreme Value, Delay

基于统计极值的流程对象时间序列时序计算算法

朱桐霖[1]，杜　涛[1]，曲守宁[1,2]，朱连江[2]

[1]济南大学信息科学与工程学院，山东 济南
[2]济南大学信息网络中心，山东 济南

文章引用: 朱桐霖, 杜涛, 曲守宁, 朱连江. 基于统计极值的流程对象时间序列时序计算算法[J]. 数据挖掘, 2016, 6(4): 179-191. http://dx.doi.org/10.12677/hjdm.2016.64020

Email: eo_dut@ujn.edu.cn

收稿日期：2016年10月9日；录用日期：2016年10月24日；发布日期：2016年10月27日

摘　要

本文针对流程对象采样数据集，提出了一种基于统计极值的流程对象环节间时序计算算法，同时通过理论分析证明了该算法的正确性。该算法通过取数据的特征点，计算环节间特征点的时间距，并通过统计方法，计算出流程对象任意两环节间的延迟时间，进而得到多环节间的时序关系。通过实际流程工业采样数据集测试，可基本准确的求得任意环节数据之间的延迟时间距以及各环节间的时序关系。

关键词

流程对象，数据挖掘，时间序列，统计，极值，延迟

1. 引言

时间序列数据是一种常见的数据表现形式，经常出现在各个领域中，例如金融、气象、工业生产、科学实验等。通过数据挖掘技术可与从大量的时间序列数据中提取出宝贵的知识用于过程优化和决策支持等方面。目前，流程工业生产中普遍应用了 DCS 分布式控制系统，该系统可实时采样流程工业生产过程中各个环节的状态值，形成时间序列数据。流程工业是由多个生产环节组成的复杂系统，系统中各环节相互影响。在流程工业中各个环节之间的影响大多是单环节依次相关的，前一环节依次影响下一环节，即前一环节的状态变化影响下一环节的状态变化，同时，这种状态的传递存在一定延迟。这种特性反映在采样数据集中表现为各个时间序列之间存在延迟相关性，或者说各环节间存在一定的时序关系。但由于某些客观因素，在采样数据集中无法直接反应出各个环节间的时序关系，而环节间的时序关系在针对流程工业等的数据挖掘中是非常重要的。

计算各环节间的时序关系主要通过计算两两环节时间序列间的延迟时间，再根据环节间的延迟时间进行时序调整得到。目前已有的时间序列延迟时间计算方法大多是基于 Pearson 提出的相关系数计算的。该方法主要通过计算两个时间序列在不同给定延迟时的相关系数，最终选取相关系数最大时的延迟作为两时间序列之间的延迟时间。这种方法可使用于不同特征的时间序列数据，并可判断出两时间序列是否具有延迟相关性，但由于需要不断尝试计算不同延迟下的相关系数，所以当数据集规模较大时计算耗费资源较多。

本文针对流程工业生产特点，考虑其采样时间序列的特征，提出了一种基于流程工业采样数据集的环节间时序计算算法，通过计算时间序列间的延迟时间获得流程工业环节间的时序关系。该算法基于统计极值的方法得到环节间的时序关系，将时间序列的极值看作是环节状态的一次变化，通过统计两时间序列各个极值点之间的延迟时间计算法出两时间序列间的延迟。算法主要包括 3 部分：1) 时间序列的极值点的计算；2) 根据极值点计算时间序列间的延迟时间；3) 根据各环节时间序列间的延迟时间得到环节间的时序关系。通过理论分析，本文证明了该算法在一定条件下的正确性。实验结果显示，对于流程工业采样数据集，该算法可准确计算出流程工业各环节间的时序关系。

本文的组织结构如下：在第 2 章中将介绍目前关于时间序列数据在延迟时间计算等方面的最新研究进展。在第 3 章中，给出流程对象的相关定义。在第 4 章中将详细介绍算法。在第 5 章将通过理论分析

证明算法的正确性。在第 6 章中将通过例子对算法进行分析。在第 7 章将以流程工业采样数据集测试算法。最后在第 8 章将对全文进行总结。

2. 相关研究

目前已有的计算时间序列的延迟时间的算法主要基于 Pearson 相关性系数计算。通过不断计算两时间序列在不同延迟下的相关性系数，取使两序列间相关性系数最大的延迟作为两个时间序列之间的延迟时间。由于该算法在计算时需要重复搜索整个时间序列，时间复杂度较高。许多学者基于相关系数计算算法提出了改进算法，通过加快搜索速度或转换时间序列形式等方法加速计算过程。Sakurai Y 等为了加速对时间序列的搜索，提出了 BRAID 算法[1]，通过引入几何渐进探查法和近似平滑来加速相关性计算。该算法通过几何渐进法减少尝试的延迟值，通过平滑时间序列减少时间序列长度加速相关系数的计算。林子雨等提出了三点预测探查法[2]，通过设置最优起始延迟，减少搜索范围加速计算。该算法根据总结的延迟值分布特点，设置三个探查点确定待搜索延迟的最优起始值，大大减少了不同延迟的尝试次数。Yue D 等提出了基于布尔序列的延迟相关性算法[3]，通过将原时间序列数据转化为布尔数据序列加速延迟计算，对于布尔序列仅通过异或运算即可得到两序列的相关系数，可加速相关系数的计算。同时 Zhang T 等在数据流挖掘算法[4]，Fungwacharakorn W 等在水文数据延迟计算[5]中都采用了类似转化布尔序列的延迟计算算法，体现了这类算法的适用性。

对于时间序列延迟时间的计算，也可将时间序列分割为多个小段，通过对各个分段进行相似性匹配，计算两时间序列之间的延迟时间。目前，时间序列的距离度量和相似性匹配是时间序列研究的热门方向，有许多新的研究成果[6] [7] [8]。Serra J 等提出了一种基于累积跳跃代价的相似性距离度量 MJC [9]。该算法通过在两个序列间按照一定规则不断跳跃，并累积从一个序列到另一个序列的跳跃代价来度量两个时间序列之间的距离。Stefan A 等提出了新的相似性度量标准 MSM [10]，该算法事先规定了 Move、Split、Merge 三种操作，并为每种操作设定了代价值，如果一个时间序列通过这三种操作转变为另一个序列，则两个序列之间的距离就可以用转换所经过的所有操作的代价和来度量。Nakamura T 等提出了一种基于形状的相似性计算方法[11]，将原来的由时间点组成的时间序列转化为由方向向量组成的序列，并使用向量夹角余弦值作为相似性度量。Boucheham B 等提出了基于约简后数据的相似性计算方法[12]，该算法首先对原时序数据进行约简然后再使用 DTW 距离等进行相似性计算，由于缩减了时间序列长度，因此可加快计算速度。H Li 等同样基于 DTW 距离并结合分段线性近似计算时间序列间的相似性[13]。丁永伟等提出了基于弧度距离的相似性度量[14]，将时间序列转化为弧度序列，通过定义的弧度序列相似性度量，计算两时间序列之间的相似性。肖瑞等提出了基于趋势的相似性度量[15]，将原时间序列分段，然后判断每个分段的曲线发展趋势，将原时间序列转变为由各段趋势组成的趋势序列，然后计算相似性。

目前对于时间序列数据的研究大多集中于时间序列的相关性研究方面，专门针对流程工业数据集的时序发现算法相对较少。原有的基于相关系数的计算方法由于需要重复计算时间序列间的相关系数，因此时间复杂度较高，多用于增量更新式计算时间序列数据流之间的延迟，对于历史大数据集的计算效率不高。而基于它的一些改进加速算法则存在计算精度不足的问题。

3. 相关定义

本文提出的时序计算算法基于流程工业采样数据集。流程工业生产过程复杂，各部分的生产环节相互影响，在对流程工业进行数据挖掘时不仅要考虑每个环节还要综合考虑各环节之间的影响关系。因此，我们通常将流程工业生产过程看作是一个整体，将其抽象为一个流程对象。流程对象由多个环节组成，每个环节由其在不同时刻的采样数据组成。通常流程对象具有如下几个特点：1) 各环节单向相关，即在

流程对象中，前一个环节产生的变化，会导致下一环节也产生变化。2) 前一环节的变化不能立即影响下一环节，而是存在一定延迟。下面，我们对流程对象作如下形式化定义。

定义 1 (流程对象)：设流程对象共有 n 个环节，每个环节包含一个或多个采样点，$X_i, i \in (1,n)$ 表示每个采样点，系统统一采样周期为 T，$t_i, i \in (1,m)$ 为采样时间，$T = t_{i+1} - t_i$，则流程对象可定义为：

$$\begin{aligned} \mathrm{X} = \{ & X_1\left(x_1(t_1), x_1(t_2), \cdots, x_1(t_m)\right), \\ & X_2\left(x_2(t_1), x_2(t_2), \cdots, x_2(t_m)\right), \\ & X_n\left(x_n(t_1), x_n(t_2), \cdots, x_n(t_m)\right)\} \end{aligned}$$

其中，$x_i(t_m)$ 表示第 i 环节在 t_m 时刻的采样值[16]。

流程对象各环节状态的传播存在延迟，对于任意两环节之间的延迟我们有如下定义。

定义 2 (延迟)：对于流程对象 X 中的任意两个环节 X_i 和 X_j，环节 X_i 的某个变化导致环节 X_j 产生某个响应变化的延迟时间记为 Δt_{ij}，则有 $\Delta t_{ij} = t(X_j) - t(X_i)$，其中 $t(X_i)$ 和 $t(X_j)$ 分别表示 X_i 环节产生变化的时刻以及这个变化传递到 X_i 环节产生响应的时刻。

4. 算法设计

4.1. 算法基本思想

流程对象的各个环节之间是单向相关的且环节间的状态传递存在延迟，也就是说其各个环节之间存在时序关系。流程工业控制系统的采样数据形式如表 1 所示。通常在采样数据库中的时间序列数据不能直接反应出流程对象各环节之间的时序关系，而流程对象环节间的时序关系是流程对象数据挖掘的重要信息。因此，我们将通过计算各环节采样时间序列间的延迟时间得到流程对象各个环节间的时序关系。

对于流程对象的一个环节，当采样周期足够小时，采样系统的采样数据足以反应出环节中状态的变化。其在平稳状态下的一次波动在采样数据中可看作是采样数据的一次较大幅度的变化。对于流程工业大数据集，每个环节的历史采样数据集非常庞大，若对这些数据集直接进行计算，计算效率较低。为提高算法效率，需要提取时间序列的特征点简化时间序列。因此，本文考虑采用时间序列中的极值点来代表流程对象一个环节状态的波动，将一个环节采样时间序列的一个极值点看作是该环节的一次波动。则前一环节的状态变化传递到下一环节表现在采样数据上就是前一环节出现一个极值点，在下一环节一段时间之后通常会有一个相应的响应极值点，如图 1 所示，每当一个环节出现一次波动，另一环节就会有一个相应相似的波动。前一环节极值点与下一环节对应的响应极值点之间的时间差值就是两个波动传播的延迟时间。在理想情况下，对应极值点之间的延迟时间就是两时间序列之间的延迟时间。但在实际情况中，通常存在采样误差以及扰动等可能会对采样数据产生干扰的情况。为了排除这些干扰的影响，得到准确的时间序列间延迟，我们在此采用了统计众数的方式——通过统计极值点之间的迟时间距的众数去除扰动点的影响得到两环节采样时间序列之间的延迟时间。

本文提出的时序发现算法主要分为两个部分。第一部分，对时间序列取局部极值，构成极值点序列。第二部分，求任意两时间序列各个极值点之间的时间距，并据此进一步求得任意时间序列之间的延迟时间。最终，我们便可根据时间序列之间的延迟时间得到各个环节采样点之间的时序关系。下面我们将分别详细介绍该算法的两个主要部分。

4.2. 时间序列极值

对流程对象中一个环节采样点的采样数据时间序列的极值有如下定义：

Table 1. Sampling data tables of the process object
表 1. 流程对象采样数据表

检测时间	环节 1	环节 2	环节 3	…	环节 n-1	环节 n
$\mathbf{T_1}$	X_{11}	X_{21}	X_{31}	…	$X_{n-1,1}$	X_{n1}
$\mathbf{T_2}$	X_{12}	X_{22}	X_{32}	…	$X_{n-1,2}$	X_{n2}
$\mathbf{T_3}$	X_{13}	X_{23}	X_{33}	…	$X_{n-1,3}$	X_{n3}
…	…	…	…	…	…	…
$\mathbf{T_m}$	X_{1m}	X_{2m}	X_{3m}	…	$X_{n-1,m}$	X_{nm}

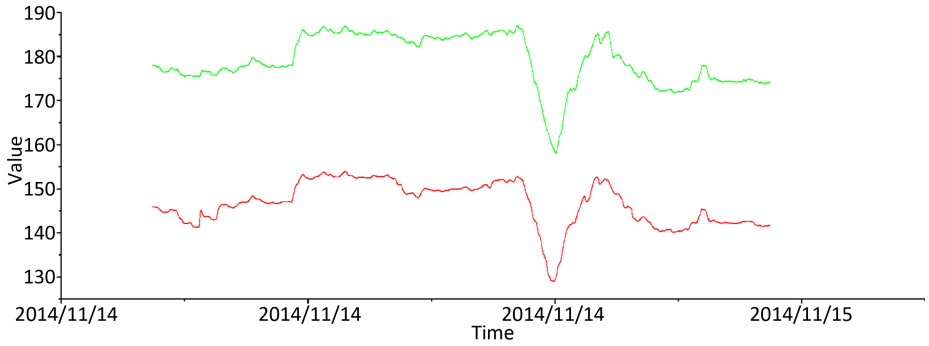

Figure 1. The sampling time series data fragment of two links
图 1. 两个环节采样数据片段

定义 3 (时间序列极值): 对于流程对象 X 的一个环节的采样时间序列 X_i，$\exists t_m \in (t_p, t_q)$，若 $x_i(t_m) > x_i(t_j)$ 或 $x_i(t_m) < x_i(t_j)$，$t_j \in [t_p, t_m) \cup (t_m, t_q]$，则 $x(t_m)$ 为环节 X_i 采样时间序列在时间段 $[t_p, t_q]$ 内的极值。

在算法中，我们首先通过计算差分的方法得到时间序列的极值。通过比较时间序列的差分的变化情况判断时间序列的变化趋势。对于时间序列的差分我们有如下定义：

定义 4 (差分): 对于时间序列 $X_i = \{x_i(t_1), x_i(t_2), x_i(t_3), \cdots, x_i(t_m)\}$，其在任意时刻 t_j 的采样值为 $x_i(t_j)$，则 $\Delta x_i(t_j) = x_i(t_{j+1}) - x_i(t_j)$ 为 X_i 的前向差分。

对于 X_i 环节时间序列 $X_i = \{x_i(t_1), x_i(t_2), x_i(t_3), \cdots, x_i(t_m)\}$，根据定义 4 对时间序列 X_i 通过后项减前项的方式逐个计算其在 $t_j \in [t_1, t_{m-1}]$ 时刻的差分值，生成差分序列 $\Delta X_i = \{\Delta x_i(t_1), \Delta x_i(t_2), \Delta x_i(t_3), \cdots, \Delta x_i(t_{m-1})\}$。然后，根据差分序列中差分的变化情况判断时间序列中的极值点。根据差分值判断极值点的方法如下：

1) 若 $\Delta x_i(t_j) > 0$ 且 $\Delta x_i(t_{j+1}) < 0$ 或 $\Delta x_i(t_{j+1}) = 0$，则 $x_i(t_j)$ 为时间序列 X_i 的在时刻 t_j 的极值。

2) 若 $\Delta x_i(t_j) < 0$ 且 $\Delta x_i(t_{j+1}) > 0$ 或 $\Delta x_i(t_{j+1}) = 0$，则 $x_i(t_j)$ 为时间序列 X_i 的在时刻 t_j 的极值。

3) 若 $\Delta x_i(t_j) = 0$ 且 $\Delta x_i(t_{j+1}) > 0$ 或 $\Delta x_i(t_{j+1}) < 0$，则 $x_i(t_j)$ 为时间序列 X_i 的在时刻 t_j 的极值。

根据以上极值点的判断方法，搜索整个时间序列 X_i 对应的差分序列 ΔX_i 找到所有极值点，得到时间序列 X_i 对应的包括极值点时刻信息的极值点序列 $X_i' = \{x_i(t_1'), x_i(t_2'), x_i(t_3'), \cdots, x_i(t_p')\}, p < m$。对于流程对象中的其它环节通过同样的过程得到对应的极值点序列。该算法的具体描述如下：

算法 1： 时间序列极值计算算法

输入：流程对象任意环节 X_i 的采样时间序列数据 $X_i = \{x_i(t_1), x_i(t_2), x_i(t_3), \cdots, x_i(t_m)\}$；

输出：环节 X_i 时间序列对应的极值点序列 X_i'。

for $j = 1$ to $j = m - 1$

$\Delta x_i\left(t_j\right)=x_i\left(t_{j+1}\right)-x_i\left(t_j\right)$；

$\Delta x_i\left(t_j\right)\to\Delta X_i$；//差分序列 $\Delta X_i=\left\{\Delta x_i\left(t_1\right),\Delta x_i\left(t_2\right),\Delta x_i\left(t_3\right),\cdots,\Delta x_i\left(t_{m-1}\right)\right\}$

endfor

for $j=1$ to $j=m-1$

if $\Delta x_i\left(t_j\right)>0$ and $\Delta x_i\left(t_{j+1}\right)<0$ or $\Delta x_i\left(t_{j+1}\right)=0$

$x_i\left(t_j\right)\to X_i'$；//记录当前时刻 t_j，$x_i\left(t_j\right)$ 写入极值序列 X_i'

if $\Delta x_i\left(t_j\right)<0$ and $\Delta x_i\left(t_{j+1}\right)>0$ or $\Delta x_i\left(t_{j+1}\right)=0$

$x_i\left(t_j\right)\to X_i'$；

if $\Delta x_i\left(t_j\right)=0$ and $\Delta x_i\left(t_{j+1}\right)<0$ or $\Delta x_i\left(t_{j+1}\right)>0$

$x_i\left(t_j\right)\to X_i'$；

//最终获得极值序列 $X_i'=\left\{x_i\left(t_1'\right),x_i\left(t_2'\right),x_i\left(t_3'\right),\cdots,x_i\left(t_p'\right)\right\}$

endfor

4.3. 时间距计算

对于流程对象中任意两个测点的时间序列 X_i 和 X_j，通过上述过程求得它们对应的极值序列分别为 $X_i'=\left\{x_i\left(t_{i1}'\right),x_i\left(t_{i2}'\right),x_i\left(t_{i3}'\right),\cdots,x_i\left(t_{ip}'\right)\right\}$ 和 $X_j'=\left\{x_j\left(t_{j1}'\right),x_j\left(t_{j2}'\right),x_j\left(t_{j3}'\right),\cdots,x_j\left(t_{jq}'\right)\right\}$，其中 $t_{ia}',a\in[1,p]$ 和 $t_{jb}',b\in[1,q]$ 分别表示两极值序列极值点对应的时刻。我们对两个时间序列不同位置极值点间的时间距有如下定义：

定义 6 (极值点时间距)：测点 X_i 在位置 a 处极值点的时刻为 t_{ia}'，测点 X_j 在位置 b 处的时刻为 t_{jb}'，则两个时间序列不同位置极值点之间的时间距为 $\Delta t_{ab}=\left|t_{ia}'-t_{jb}'\right|$。

计算环节 X_j 相对于环节 X_i 的时间距时分为两部分。第一部分计算两环节不同位置极值点间的时间距，第二部取时间距最小值的众数。

对于计算极值点时间距部分。首先，我们以环节 X_i 为基环节，根据定义 6 计算其在位置 $a,a\in[1,p]$ 处极值点与环节 X_j 各个位置极值点的时间距。得到环节 X_i 在 a 位置极值点 $x_j\left(a'\right)$ 对于环节 X_j 各极值点时间距的结果集，记为 $\chi_a=\left\{\Delta t_{a1},\Delta t_{a2},\cdots,\Delta t_{aq}\right\}$。从结果集中取出最小值，记为 τ_a。其次，对基环节 X_i 的其它各极值点分别进行上述计算，每次记录其中的最小值，得到测点 X_i 各极值点对于测点 X_j 所有极值点距离的最小值集合，记为 $\mathrm{T}=\left\{\tau_1,\tau_2,\cdots,\tau_p\right\}$。

对于计算众数部分，我们首先对上述计算过程得到的最小值集合 $\mathrm{T}=\left\{\tau_1,\tau_2,\cdots,\tau_p\right\}$ 进行排序得到排序后的集合 T′，然后搜索有序集合 T′，统计每个最小值出现的次数，得到集合的众数，记为 M_{ij}。则测点 X_i 与 X_j 之间的延迟时间距 $\Delta t_{ij}=M_{ij}$。

算法 2：时间序列间延迟时间距计算算法

输入：X_i 和 X_j 对应的极值序列 X_i' 和 X_j'；

输出：两时间序列间的延迟时间距 Δt_{ij}。

for a=1 to a=p

for b=1 to b=q

$\Delta t_{ab}=\left|t_{ia}'-t_{jb}'\right|$;

$\Delta t_{ab}\to\chi_a$；//存入集合 χ_a，最终得到集合 $\chi_a=\left\{\Delta t_{a1},\Delta t_{a2},\cdots,\Delta t_{aq}\right\}$

endfor

$\tau_a = \Delta t_{a1}$;

for b=2 to b=q

if $\left(\Delta t_{ab} - \Delta t_{a(b-1)}\right) < 0$

$\tau_a = \Delta t_{ab}$;

$\tau_a \to \mathrm{T}$;//存入集合T，最终得到集合$\mathrm{T} = \left\{\tau_1, \tau_2, \cdots, \tau_p\right\}$

endfor

endfor

$M_{ij} = \mathrm{MODE}(\mathrm{T})$//对集合T取众数

$\Delta t_{ij} = M_{ij}$//得到两时间序列间的时间距

4.4. 时序计算

以流程对象中第一个环节X_1为基环节，通过上述极值计算和时间距计算两步分别计算其与流程对象中其它环节$X_2 \cdots X_n$之间的延迟时间，然后将各环节间的延迟时间由大到小进行排序，进而便可根据延迟时间的顺序得到流程对象各环节间的时序关系。

5. 理论分析

在流程工业生产过程中，各环节间状态变化传递的都是连续信号，即一个环节的输入和输出都是连续信号。而在流程工业控制系统中通过施加脉冲信号得到的采样结果为离散序列。在流程工业中一个生产环节通常可看作由一个纯滞后环节和一个非滞后环节串联组成。为了便于讨论，我们在此认为流程对象中一个环节X_i由G_1和G_2两个环节串联组成。如图2所示，$R(s)$为X_i输入信号即前一环节X_{i-1}的输出信号的拉氏变换，$C(s)$为环节X_i输出信号的拉氏变换。则根据计算机控制系统理论，我们可以有如下定理。

定理1：流程对象两相邻环节采样时间序列之间，其采样值存在延迟相关关系。

证明：

设第一个环节的输出为$c_1(t)$，其拉氏变换为$C_1(s)$，则有

$$C_1(s) = G_1(s)R^*(s)$$

第二个环节的输出为连续时间函数$c(t)$，其拉氏变换为$C(s)$，则有

$$C(s) = G_2(s)C_1(s) = G_2(s)G_1(s)R^*(s)$$

由于连续信号经过采样开关后变为离散信号，故对上式两边同时离散化，得

$$C^*(s) = \left[G_2(s)G_1(s)R^*(s)\right]^* = \left[G_2(s)G_1(s)\right]^* R^*(s)$$

对上式作z变换，令$s = T^{-1}\ln z$则有

$$C(z) = G_2G_1(z)R(z)$$

对上式进行z反变换，则有

$$c(kT) = Z^{-1}\left[G_2G_1(z)\right]r(kT)$$

其中，T为采样周期。由上式可得流程对象前一环节与后一环节之间存在随采样时间变化的函数，因此两环节采样时间序列间存在延迟相关关系。

证毕

通过定理1我们可以知道流程对象中任意相邻两环节的采样时间序列间存在函数关系$X_i(kT) = f\left(X_{i-1}(kT)\right)$，

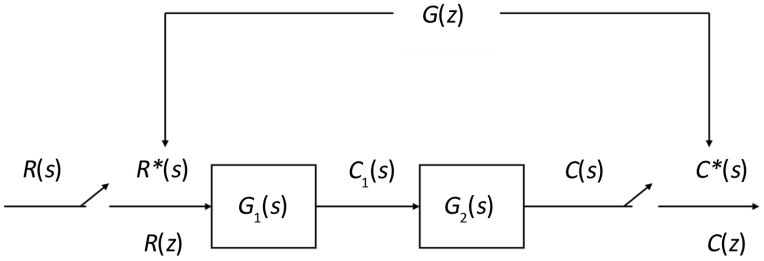

Figure 2. A link in the process object consists of G_1 and G_2
图 2. 流程对象中一个环节

将其推广到任意两环节之间我们可以有以下推论。

推论 1：流程对象中任意两环节的采样时间序列间，存在延迟相关关系。

根据定理 1 和推论 1 可以知道流程对象任意两环节间的采样时间序列间存在延迟相关关系，因此我们便可以通过时间序列各个采样值的时间距计算环节间的延迟时间。关于环节间延迟时间的计算我们有如下定理。

定理 2：采样点 X_i 与 X_j 的时间序列对应的极值点序列为 $X_i' = \left\{x_i\left(t_1'\right), x_i\left(t_2'\right), x_i\left(t_3'\right), \cdots, x_i\left(t_p'\right)\right\}$ 与 $X_j' = \left\{x_j\left(t_1'\right), x_j\left(t_2'\right), x_j\left(t_3'\right), \cdots, x_j\left(t_q'\right)\right\}$。系统采样周期为 T，环节内极值点出现的最小间隔为 $t_{\min}$，$t_{\min} = \min\left|x_i\left(t_{a+1}'\right) - x_i\left(t_a'\right)\right|, a \in (1, p)$。在不存在扰动的理想情况下，设两环节之间的延迟时间为 Δt，则当 $\Delta t < \frac{1}{2}t_{\min}$ 时，采样点 X_i 在某一位置 a 的极值点与其在环节 X_j 对应 a' 位置的响应极值点的延迟时间总是 X_i 位置 a 到 X_j 任意位置距离的绝对值最小的。

证明：

当采样点 X_i 在 X_j 之前时，对于采样点 X_i 时间序列中的一个极值点 $x_i\left(t_a'\right), a \in [1, p]$，$X_j$ 中有其对应的响应极值点。显然，$x_i\left(t_a'\right)$ 距 X_j 各极值点时间距的最小值一定是 $x_j\left(t_a'\right)$ 和 $x_j\left(t_{a-1}'\right)$ 之一，其中 $x_j\left(t_a'\right)$ 即为对应的响应极值点。极值点 $x_i\left(t_a'\right)$ 到 $x_j\left(t_a'\right)$ 和 $x_j\left(t_{a+1}'\right)$ 的时间距分别为：$d_{a(a-1)} = t_{ai}' - t_{(a-1)j}'$

$$d_{aa} = t_{ai}' - t_{aj}'$$

其中，c 表示极值点 $x_i\left(t_a'\right)$ 发生的时刻。因为 $\left(t_{ai}' - t_{(a-1)j}'\right) \geq t_{\min}$，$\Delta t < \frac{1}{2}t_{\min}$，而 $\Delta t = d_{aa} = t_{ai}' - t_{aj}'$。因此可得到 $\left(t_{ai}' - t_{(a-1)j}'\right) > \frac{1}{2}t_{\min} > t_{ai}' - t_{aj}'$。因此当 $\Delta t < \frac{1}{2}t_{\min}$ 时，采样点 X_i 在某一位置 a 的极值点与其在环节 X_j 对应 a' 位置的响应极值点的延迟时间一定是 X_i 位置 a 到 X_j 任意位置距离的绝对值最小的。

当采样点 X_i 在 X_j 之后时，同理可证。

证毕

推论 2：在没有扰动的理想情况下，采样点 X_i 与 X_j 之间的延迟时间距为 X_i 中各极值点与 X_j 中对应极值点的时间距。

推论 3：当存在扰动且扰动点较少时，采样点 X_i 与 X_j 之间的延迟时间距为 X_i 中各极值点与 X_j 中对应极值点的时间距最小值相同数目最多的。

通过以上定理以及推论我们可以确定在流程对象采样数据集上，本文提出的算法是正确的，可以通过该算法计算出流程对象各环节间的时序关系。

6. 算法分析

在上一节中我们通过理论分析论证了算法的正确性，在本节我们将通过三个例子来分析算法在不同

情况下的适用情况。

在流程工业采样系统中，存在采样周期，即每次采样之间有一定的时间间隔。经分析，流程对象的采样周期、各环节中相邻每次波动产生的最短时间间隔等因素会影响该算法的准确性。由于流程对象采样系统可能存在的故障误差等因素，每个环节的每次波动不一定全部被采样到，反应在采样数据中表示为时间序列极值点的缺失。针对以上影响因素，我们分别讨论算法对不同情况的处理。

对于 X_i 与 X_j 两个环节， X_j 与 X_i 的延迟时间距，设为 Δt。系统采样周期为 T，各个环节中每次波动产生的最小时间间隔为 $t_{\min}$。

由图 3 我们可以看到，$\Delta t < \frac{1}{2}t_{\min}$ 且采样数据不存在扰动时，环节 X_i 的一个极值点到环节 X_j 各个极值点的时间距最小值一定是其与对应的响应极值点间的时间距，这个最小时间距也是两环节之间的延迟时间。以环节 X_i 的极值点 b 与对应的环节 X_j 的极值点 b' 为例，b 与 b' 之间的时间距为 $\left|t(b')-t(b)\right|$，b 与 a' 之间的时间距为 $\left|t(a')-t(b)\right|$，显然 $\left|t(b')-t(b)\right| < \left|t(a')-t(b)\right|$。

而对于图 4 所示的存在扰动的情况，对于环节 X_i 中的极值点 d，其在环节 X_j 中对应的响应极值点没有被采到。此时 d 极值点与环节 X_j 各极值点时间距的最小值就不是其与对应响应极值点的时间距。算法中的统计众数方法便是为了解决这种情况的。当这种扰动没有干扰到绝大多数数据点时，正常的有对应响应极值点的总是占大多数的。因此，通过统计众数的方法便可去除这些扰动点的干扰，得到环节间的延迟时间。

在图 5 中，$\Delta t > \frac{1}{2}t_{\min}$，并且存在一个扰动点。对于 b 极值点，其距离前一个极值点 a 的时间距

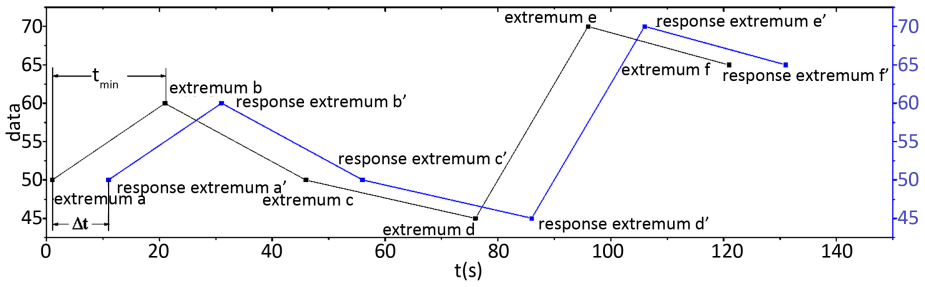

Figure 3. $\Delta t < \frac{1}{2}t_{\min}$

图 3. $\Delta t < \frac{1}{2}t_{\min}$

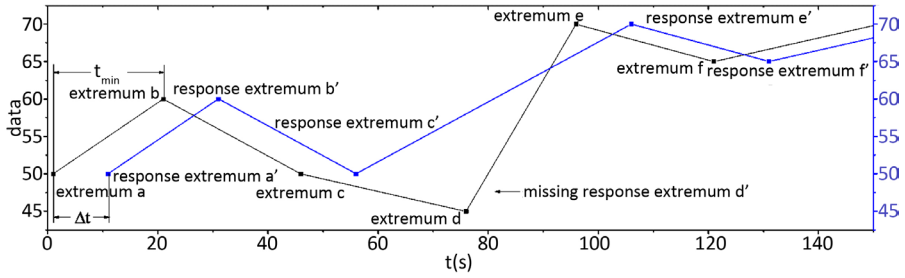

Figure 4. When $\Delta t < \frac{1}{2}t_{\min}$ and existence disturbance

图 4. $\Delta t < \frac{1}{2}t_{\min}$ 且存在扰动

$t = t_{\min} < 2\Delta t$，显然$\left|t(b') - t(b)\right| > \left|t(a') - t(b)\right|$，此时环节$X_i$的$b$极值点与环节$X_j$各极值点时间距的最小值不是$b$与对应极值点$b'$之间的时间距，也必然不是两环节间的延迟时间。但当环节中极值点距离$t < 2\Delta t$出现的次数较少且扰动点出现的次数较少时，该算法仍然可以通过统计众数的方式，去除这些情况的干扰。当去掉图 5 中的扰动点时，就变为$\Delta t > \frac{1}{2}t_{\min}$，且不存在扰动点的情况，如图 6 所示。根据上述分析，当环节中极值点距离$t < 2\Delta t$出现的次数较少时，该算法同样可以通过统计众数的方式，求得准确的延迟时间距。

通过以上几种情况的分析，我们可以看到本文提出的算法具有较强的抗干扰能力，在下一章将通过实验来检测算法的效果。

7. 实验

为了检测本文提出的算法应用于时序计算时的效果，我们在此使用流程工业采样数据作为测试数据集。本文的时序计算算法通过计算时间序列之间的延迟时间来得到环节间的时序关系的，因此，此处我们仅使用流程对象采样数据集中的两个环节来进行试验，使用该算法计算两个环节采样时间序列之间的延迟时间。如图 7 所示，为我们所使用的两个环节采样数据，每个环节的时间序列包括 14,997 个采样值，该组数据的采样周期为 60 s。红色曲线表示环节X_1的采样时间序列，绿色曲线表示环节X_2的采样时间序列。从图中可以看到，一个环节产生波动另一个环节会有一个对应的响应波动，两个时间序列的波形相似，且存在一定的延迟。

根据本文中的算法，首先计算的是两个时间序列的差分，分别得到两个时间序列的差分序列。然后在差分序列的基础上计算两个时间序列的极值，通过计算我们可以得到两个环节采样时间序列对应的极

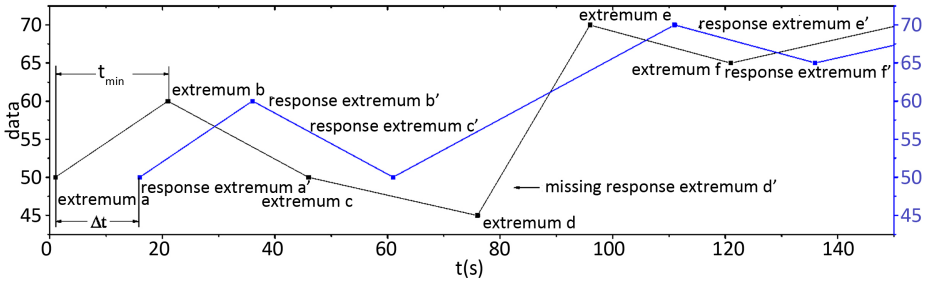

Figure 5. When $\Delta t > \frac{1}{2}t_{\min}$ and existence disturbance

图 5. $\Delta t > \frac{1}{2}t_{\min}$且存在扰动

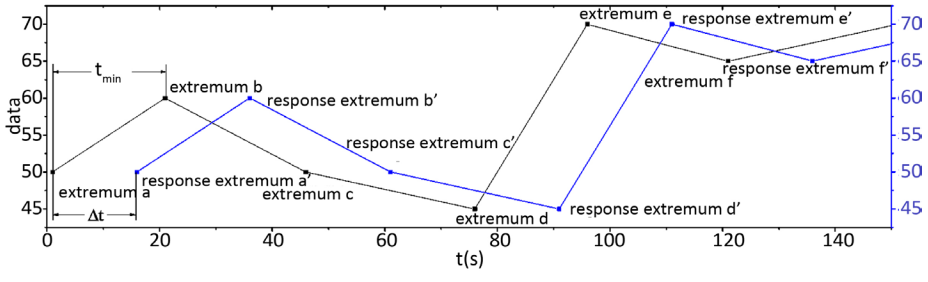

Figure 6. $\Delta t > \frac{1}{2}t_{\min}$

图 6. $\Delta t > \frac{1}{2}t_{\min}$

值点序列。在图 8 中我们可以看到两个时间序列中极值点的分布情况。其中，红色点为环节 X_1 的采样时间序列的极值点，绿色点为环节 X_j 的采样时间序列极值点。

进一步地，我们对环节 X_1 的每个极值点计算其与环节 X_j 的各个极值点之间的时间距，然后取其中的最小值。统计这些极值点间的最小时间距，得到两环节各个极值点间最小时间距的分布情况，如图 9 所示。从图中可以看到延迟为−60 s 在分布中占大多数，因此，本次试验所选取的两个环节时间序列之间的延迟时间为−60 s，而系统采样周期为 60 s，即环节 X_1 在环节 X_2 之前一个采样周期的位置。这就得到了流程对象其中两个环节间的时序关系，同样，对于其他环节间的时序关系可通过此算法依次得到。

我们对试验结果做进一步分析，从图 9 中我们可以看到，除去延迟为−60 s 占据大多数，在分布图中也可以看到 0 s 以及−120 s 等的延迟也有许多。而在不存在干扰的理想情况下，环节中一个极值点与另一环节所有极值点时间距最小值应该为该极值点与另一环节对应响应极值点的时间距，即两环节间的延迟时间。但在如本节试验所取的实际生产采样数据中，存在大量的采样误差和扰动等干扰情况，在这种干

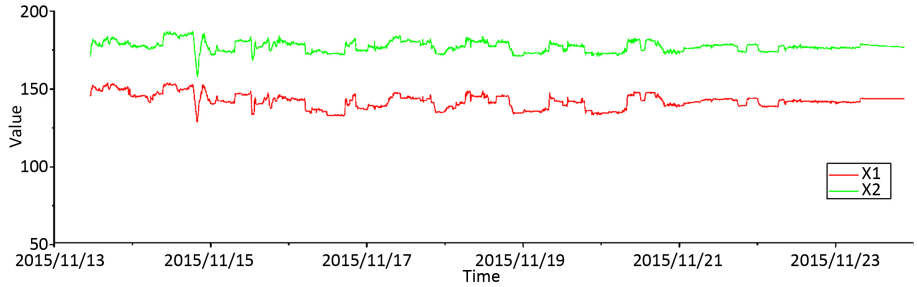

Figure 7. The original sampling time series
图 7. 原采样时间序列

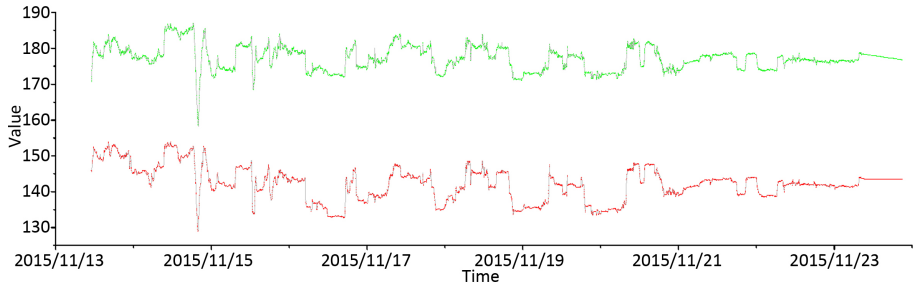

Figure 8. Extreme points sequence
图 8. 极值序列

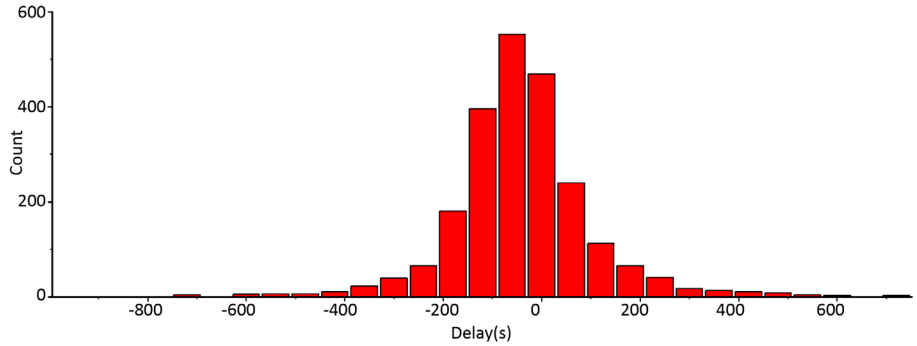

Figure 9. Minimum time distance distribution between the extreme points
图 9. 极值点间最小时间距分布

扰的情况下，环节中一个极值点与另一环节所有极值点时间距最小值不一定为该极值点与对应响应极值点间的时间距。另一方面，对于时间序列中没有受到干扰的极值点，该极值点与另一环节各极值点时间距的最小值仍然为与其对应的响应极值点间的时间距，即环节间的延迟时间。因此，当这种干扰没有影响到时间序列中绝大多数采样值时，便可通过统计众数的方法，取得最多的时间距作为环节间的延迟时间。可以在存在干扰的情况下准确计算出各环节采样时间序列间的延迟时间。

8. 总结

时间序列处理一直是数据挖掘领域的研究热点。时间序列间延迟计算在许多领域中都有应用。特别的，对于流程工业数据挖掘的时序计算过程，就是通过计算时间序列间的延迟时间得到的。本文提出的时序计算算法在时间序列延迟计算过程中，充分考虑了流程工业的生产状态变化情况，根据其时间序列数据的特征，提出的以极值点作为特征点计算时间距的算法可有效计算出环节间的时间延迟，同时通过统计众数的方法有效克服了实际生产数据中存在的大量干扰对计算结果造成的干扰。

另一方面，由于该算法是在流程对象数据集的基础上提出的，对于不同的时间序列数据可能存在泛用性问题。下一步我们考虑使用时间序列间的模式匹配方法计算时间序列延迟时间，以进一步增加算法的准确性，同时提高算法针对不同情况的时间序列数据的泛用性。

参考文献 (References)

[1] Sakurai, Y., Papadimitriou, S. and Faloutsos, C. (2005) Braid: Stream Mining through Group Lag Correlations. *Proceedings of the* 2005 *ACM SIGMOD International Conference on Management of Data*, Baltimore, 14-16 June 2005, 599-610. http://dx.doi.org/10.1145/1066157.1066226

[2] 林子雨, 江弋, 赖永炫, 林琛. 一种新的时间序列延迟相关性分析算法——三点预测探查法[J]. 计算机研究与发展, 2012(12): 2645-2655.

[3] Yue, D., Zhang, T., Yu, G., *et al.* (2007) Lag Correlation Analysis Based on Boolean Presentation over Multiple Data Streams. International Conference on Intelligent Systems and Knowledge Engineering. Atlantis Press, Paris. http://dx.doi.org/10.2991/iske.2007.133

[4] Zhang, T., Yue, D., Wang, Y., *et al.* (2011) A Novel Approach for Mining Multiple Data Streams Based on Lag Correlation. 2011 *Chinese Control and Decision Conference* (*CCDC*), Mianyang, 23-25 May 2011, 2377-2382. http://dx.doi.org/10.1109/CCDC.2011.5968606

[5] Fungwacharakorn, W. and Pattara-Atikom, W. (2014) Enhancement of Lag Time Query on Hydrologic Data Using Clipping Technique and Logic-Based Correlation. 2014 11*th International Conference on Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology* (*ECTI-CON*), Nakhon Ratchasima, 14-17 May 2014, 1-6. http://dx.doi.org/10.1109/ECTICon.2014.6839813

[6] 武红江, 赵军平, 彭勤科, 黄永宣. 基于波动特征的时间序列数据挖掘[J]. 控制与决策, 2007, 22(2): 160-163.

[7] 谢福鼎, 王赫楠, 张永. 一种新的时间序列线性拟合方法[J]. 计算机工程, 2011, 37(22): 250-251+254.

[8] 李海林. 基于动态弯曲的时间序列异步相关性分析[J]. 计算机应用研究, 2014, 31(7): 1976-1979.

[9] Serra, J. and Arcos, J.L. (2012) A Competitive Measure to Assess the Similarity between Two Time Series. Case-Based Reasoning Research and Development. Springer, Berlin Heidelberg, 414-427. http://dx.doi.org/10.1007/978-3-642-32986-9_31

[10] Stefan, A., Athitsos, V. and Das, G. (2013) The Move-Split-Merge Metric for Time Series. *IEEE Transactions on Knowledge and Data Engineering*, **25**, 1425-1438. http://dx.doi.org/10.1109/TKDE.2012.88

[11] Nakamura, T., Taki, K., Nomiya, H., *et al.* (2013) A Shape-Based Similarity Measure for Time Series Data with Ensemble Learning. *Pattern Analysis and Applications*, **16**, 535-548. http://dx.doi.org/10.1007/s10044-011-0262-6

[12] Boucheham, B. (2010) Reduced Data Similarity-Based Matching for Time Series Patterns Alignment. *Pattern Recognition Letters*, **31**, 629-638. http://dx.doi.org/10.1016/j.patrec.2009.11.019

[13] Li, H., Guo, C. and Qiu, W. (2011) Similarity Measure Based on Piecewise Linear Approximation and Derivative Dynamic Time Warping for Time Series Mining. *Expert Systems with Applications*, **38**, 14732-14743. http://dx.doi.org/10.1016/j.eswa.2011.05.007

[14] 丁永伟, 杨小虎, 陈根才, Kavs, A.J. 基于弧度距离的时间序列相似度量[J]. 电子与信息学报, 2011, 33(1): 122-128.

[15] 肖瑞, 刘国华. 基于趋势的时间序列相似性度量和聚类研究[J]. 计算机应用研究, 2014, 31(9): 2600-2605.

[16] Song, Q., Guo, Q., Wang, K., *et al.* (2014) A Scheme for Mining State Association Rules of Process Object Based on Big Dat. *Journal of Computer and Communications*, **2**, 17-24. http://dx.doi.org/10.4236/jcc.2014.214002