

# 基于秩2更新的多维数据流典型相关跟踪算法

杨 静,李文平,张健沛

(哈尔滨工程大学计算机科学与技术学院,黑龙江哈尔滨 150001)

**摘 要:** 现存的多维数据流典型相关分析(Canonical Correlation Analysis,简称 CCA)算法主要是基于近似技术的求解方法,本质上并不是持续更新的精确算法.为了能在时变的环境中持续、快速而精确地跟踪数据流之间的相关性,本文提出一种多维数据流典型相关跟踪算法 TCCA.该算法基于秩2更新理论,通过并行方式持续更新样本协方差矩阵的特征子空间,进而实现多维数据流典型相关的快速跟踪.理论分析及仿真实验结果表明,TCCA 具有较好的稳定性、较高的计算效率和精度,可以作为基本工具应用于数据流相关性检测、特征融合、数据降维等数据流挖掘领域.

**关键词:** 多维数据流;典型相关分析;秩2更新;快速跟踪;特征子空间

**中图分类号:** TP391; TP311.1 **文献标识码:** A **文章编号:** 0372-2112 (2012)09-1765-10

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.3969/j.issn.0372-2112.2012.09.011

## A Tracking Algorithm Based on Rank Two Modifications for Canonical Correlation Analysis of Multidimensional Data Streams

YANG Jing, LI Wen-ping, ZHANG Jian-pei

(College of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang 150001, China)

**Abstract:** Existing algorithms for canonical correlation analysis(CCA) of multidimensional data streams are mostly based on approximate techniques, but are not the precise algorithms for updates in essence. In this study, a novel canonical correlation analysis algorithm, called TCCA(Tracking CCA), is proposed for tracking the correlations rapidly and accurately between two multidimensional data streams in the time-varying environments. By introducing the technique of rank two modifications to update the eigensubspace of the sample covariance matrix in parallel, TCCA can rapidly track the correlations of data streams. Theoretical analysis and experimental results indicate that the TCCA algorithm has better stability, high computational efficiency and accuracy. It could be presented as a basic tool for correlation detection on data streams, feature fusion, dimension reduction and other areas of data streams mining.

**Key words:** multidimensional data streams; canonical correlation analysis; rank two modifications; fast tracking; eigensubspace

## 1 引言

近年来,数据流挖掘研究已经引起了学术界广泛关注.不少应用领域产生的数据都以数据流形式出现,如网络日志数据、传感器监测数据、股票交易记录、移动对象跟踪数据以及电话通话记录等.在数据流环境中,数据持续不断到达,且新数据产生的速度较快,对这类数据无法存储全部历史记录,对其挖掘分析常需要设计单遍扫描的快速算法,这给数据流挖掘带来极大挑战.

相关分析是数据流挖掘的重要内容之一.现实应用中数据流常常显露出较强的相关性.目前关于数据流相

关分析的研究主要包括单组和两组数据流的相关性研究两个方面.前者主要研究多条单维数据流构成的组内数据流之间的相关关系,而尤以滞后相关研究最具代表性,这方面的研究成果较多,如文献[1,2]提出一种名为 BRAID 的方法,用于检测多条数据流之间的滞后相关性,并给出滞后的时间长度;文献[3]基于离散傅里叶变换(Discrete Fourier Transforms,简称 DFT)提出一种名为 StatStream 的方法,可检测多条数据流之间的相关性;文献[4]利用主成分分析(Principal Component Analysis,简称 PCA)方法,跟踪多条数据流之间的相关性以及隐藏变量;文献[5]提出一种名为 HBR 的用于分析数据流相关

收稿日期:2011-10-24;修回日期:2012-04-16

基金项目:国家自然科学基金(No.61073043, No.61073041);黑龙江省自然科学基金(No. F200901);哈尔滨市科技创新人才研究专项资金(优秀学科带头人)(No.2011RFXXG015, No.2010RFXXG002);高等学校博士学科点基金(No.20112304110011)

性的方法,可通过布尔操作获得相关集;文献[6]基于 DFT 和图分割(Graph Partitioning)技术研究一种近似计算数据流相关系数的方法;文献[7]基于多尺度分析(Multi-Scale Analysis)技术研究数据流相关性问题的。

而两组数据流之间的相关性主要研究两组多维数据流之间的相关关系.研究两随机向量之间相关性的常用方法是典型相关分析(Canonical Correlation Analysis, 简称 CCA)方法. CCA 是研究两组变量之间线性相关关系的一种多元统计方法,也是一种数据降维技术,最早由 Hotelling 于 1936 年提出<sup>[8]</sup>.该方法将两组随机变量(向量)之间的线性相关性转化为少数几对互不相关的随机变量(向量)的线性相关性,目标在于寻找两组基,使得原向量在此基上的投影的 Pearson 相关系数达到最大值.因此 CCA 不仅可直接用于相关性检测,而且可用于特征融合<sup>[9]</sup>和数据降维<sup>[10,11]</sup>等领域.

多维数据流典型相关分析属于数据流挖掘的难点问题之一,关于这方面研究的公开文献报道比较有限,文献[12~15]提出的捕捉数据流相关性的方法是目前最具代表性的研究成果.其中,文献[12]基于低阶近似(Low Rank Approximations)理论,利用奇异值分解(Singular Value Decomposition, 简称 SVD)技术,能以同步和异步两种方式检测多数据流的相关性;文献[13]采用不等概抽样技术形成低阶概要矩阵,以约减元组数量,从而减少数据流处理时间,在此基础上计算多维数据流之间的前  $k$  典型相关系数;文献[14,15]采用不等概抽样技术,基于低阶近似理论,提出适于数据流处理的多变量相关性分析的增量计算方法.尽管上述方法具有较好的效果,但本质上并不是持续更新的快速算法,而是基于近似技术的求解方法.本文侧重于面向多维数据流的典型相关分析的精确、持续更新.

本文拟研究一种用于两条多维数据流典型相关跟踪的算法,基本目标在于,以持续更新的方式跟踪数据流的相关性.基本思路是,基于滑动窗口,根据前一时刻的相关状态和当前时刻流入和流出窗口的数据,求解当前时刻的相关状态,而不是像现有方法那样通过当前时刻窗口内的所有数据重新计算相关状态.据笔者所知,目前并未发现关于这方面的研究报告.需要说明的是,尽管本文研究的算法是精确算法,但也适用于近似计算,现有很多近似技术都可采纳,包括文献[12~15]中采用的低阶近似和不等概抽样技术等.

本研究引入秩 2 更新作为多维数据流典型相关持续更新的实现手段.秩 2 更新是基于滑动窗口更新协方差矩阵的一种有效方法,它具有两大优势,即持续更新和并行求解能力.此优势正是数据流快速处理所追求的理想目标.笔者认为,在基于滑动窗口的数据流处理技术中,一个快速算法本质上应该是一个持续更新的

算法,即算法应该具有维持状态的能力,当前时刻的状态应该在前一时刻状态的基础上,仅以流入和流出窗口的数据为参数更新而得,而不是用当前时刻窗口内的所有数据重新建立状态.后者显然是低效的,因为当前时刻窗口内的数据仅有一条是新流入的,其余全是之前流入的数据.然而,现存的、为数不多的多维数据流典型相关分析的所有算法皆不具有维持状态的能力,这是本文引入秩 2 更新的根本缘由之一.此外,秩 2 更新具有的并行求解能力对于数据流快速处理具有更强的吸引力,它使得多维数据流典型相关分析在并行环境下快速求解成为可能,这是本文引入秩 2 更新的另一重要理由.

## 2 基础知识回顾

### 2.1 典型相关分析(CCA)

给定  $p$  维随机向量  $X$  和  $q$  维随机向量  $Y$ ,  $p \leq q$ , CCA 的目标是寻找投影向量  $\alpha_k$  和  $\beta_k$ ,  $k = 1, 2, \dots, p$ , 使得在方差  $\text{Var}(\alpha_k^T X) = \text{Var}(\beta_k^T Y) = 1$  的约束下, Pearson 相关系数

$$\rho(\alpha_k^T X, \beta_k^T Y) = \frac{\alpha_k^T C_{xy} \beta_k}{\sqrt{(\alpha_k^T C_{xx} \alpha_k) \times (\beta_k^T C_{yy} \beta_k)}} \quad (1)$$

达到最大值.其中,  $C_{xy} = C_{yx}^T = XY^T$  为  $X$  和  $Y$  之间的互协方差矩阵,而  $C_{xx} = XX^T$  和  $C_{yy} = YY^T$  分别为  $X$  和  $Y$  的自协方差矩阵.称  $\alpha_k^T X$  和  $\beta_k^T Y$  为  $X$  和  $Y$  的第  $k$  对典型相关变量,而其相关系数称为第  $k$  个典型相关系数.

CCA 求解实质是一个最优化问题.以第一对典型变量为例(省略  $\alpha_1$  和  $\beta_1$  的下标),即求:

$$\max_{\alpha \in \mathbb{R}^{p \times 1}, \beta \in \mathbb{R}^{q \times 1}} \alpha^T C_{xy} \beta, \quad \text{s.t. } \alpha^T C_{xx} \alpha = 1, \beta^T C_{yy} \beta = 1 \quad (2)$$

其中, s.t. 表示约束条件,  $\mathbb{R}$  为实数域.用拉格朗日乘子法(Lagrange)求解式(2)有:

$$\beta = \frac{1}{\lambda} C_{yy}^{-1} C_{yx} \alpha, \quad C_{xy} C_{yy}^{-1} C_{yx} \alpha = \lambda^2 C_{xx} \alpha \quad (3)$$

式(3)第二式是广义特征值问题,由此解出  $\lambda$  和  $\alpha$ ,代入左式可得  $\beta$ .  $\lambda$  即为所求典型相关系数. CCA 有多种解法,如基于 SVD 的方法等,具体可参阅文献[16,17].下文将以式(3)为基础.

### 2.2 秩 2 更新理论

秩 2 更新(Rank Two Modifications)最早由 DeGroat 和 Roberts 于 1985 年给出<sup>[18]</sup>.对于更新式  $C(l) = C(l-1) + a(l)[a(l)]^T - b(l)[b(l)]^T$ ,其中  $C(l)$  是第  $l$  时刻的协方差矩阵,  $a(l)$  和  $b(l)$  分别是添加和删除的  $n$  维数据向量.秩 2 更新的问题是如何基于  $C(l-1)$  的特征对  $(d, v)$  求解  $C(l)$  的特征对  $(\lambda, x)$ .

记  $C(l)$  和  $C(l-1)$  分别为  $\hat{C}$  和  $C$ ,  $a(l)$  和  $b(l)$  分别为  $a$  和  $b$ .令  $C = VDV^T$  为  $C$  的特征分解式,  $\alpha =$

$V^T \mathbf{a}, \beta = V^T \mathbf{b}$ . 考虑  $T = V^T \hat{C} V = D + \alpha \alpha^T - \beta \beta^T$ , 得  $T$  的特征多项式为  $f(\lambda) = \det(D - \lambda I) g(\lambda)$ , 其中

$$g(\lambda) = \det(I + (D - \lambda I)^{-1}(\alpha \alpha^T - \beta \beta^T))$$

$$= (1 + \sum_{i=1}^n \alpha_i^2 / (d_i - \lambda)) (1 - \sum_{i=1}^n \beta_i^2 / (d_i - \lambda)) + (\sum_{i=1}^n \alpha_i \beta_i / (d_i - \lambda))^2 \quad (4)$$

由式(4)求解关于  $\lambda$  的方程  $g(\lambda) = 0$ , 其解即为  $C(l)$  对应的特征值<sup>[18]</sup>. 显然  $\lambda$  的值由前一时刻的协方差矩阵  $C(l-1)$  对应的特征值  $d$  和添加、删除的数据向量  $\mathbf{a}(l)$  和  $\mathbf{b}(l)$  更新而得.

文献[19]总结了 DeGroat 等多位研究者关于并行求解方程  $g(\lambda) = 0$  的实现思路: 根据广义交织定理 (Generalized Interleaf Theorem), 用谱分割公式将上次的特征值分割为互不相交的间隔, 再用 Newton 法并行搜索, 可快速求出新的特征值. Newton 法是二次收敛的, 常常只需一步搜索即可找到满足精度要求的解.

特征向量根据特征值更新而得<sup>[18]</sup>. 设更新特征值仍然存入  $D$  中, 不妨记为  $D = \text{diag}(d'_1, d'_2, \dots, d'_n)$ . 首先通过并行方式求解  $\mathbf{q}_k = (\eta_k \alpha + \mu_k \beta)(D - d'_k I)^{-1}$ , 其中  $\eta_k = 1 + \beta^T (D - d'_k I)^{-1} \beta$ ,  $\mu_k = \alpha^T (D - d'_k I)^{-1} \beta$ ; 其次单位化  $\mathbf{q}_k = \mathbf{q}_k / \|\mathbf{q}_k\|$ ,  $\mathbf{q}_k$  以列向量形式组成正交矩阵  $Q$ ; 最后置矩阵  $V' = VQ$ , 其列向量即为更新后的特征向量.

显然, 持续更新和并行求解是秩 2 更新的两大优势, 这是本文引入此方法的根本原因.

### 3 研究目的及思路

本文选取滑动窗口模型作为数据流典型相关分析的处理模型. 随着时间推移, 窗口不断向前移动, 每前移一步, 就从窗口内删除一条旧数据, 并插入一条新数据, 这即是所谓的滑动窗口模型. 设宽度为  $N$  的窗口内包含维数分别为  $p$  和  $q$  的两条数据流, 记第  $l$  时刻窗口内的数据矩阵为:

$$\mathbf{Z}^{(l)} = [\mathbf{z}_1^{(l)}, \mathbf{z}_2^{(l)}, \dots, \mathbf{z}_N^{(l)}] = \begin{bmatrix} \mathbf{X}^{(l)} \\ \mathbf{Y}^{(l)} \end{bmatrix} \quad (5)$$

本研究目的在于, 随着  $l$  的增加, 持续计算数据流  $\mathbf{X}^{(l)}$  和  $\mathbf{Y}^{(l)}$  之间的典型相关系数及典型相关变量.

CCA 的求解以式(3)为基础, 用样本协方差矩阵作为总体协方差矩阵的无偏估计. 令  $S_{xx}$ 、 $S_{xy}$ 、 $S_{yx}$  和  $S_{yy}$  是与总体协方差分块矩阵  $C_{xx}$ 、 $C_{xy}$ 、 $C_{yx}$  和  $C_{yy}$  分别对应的样本协方差分块矩阵, 于是得(3)右式的对应式:

$$A\alpha = \mu B\alpha \quad (6)$$

其中  $A = S_{xy} S_{yy}^{-1} S_{yx}$ ,  $B = S_{xx}$ ,  $\mu = \lambda^2$ . 显然式(6)是一个广义特征值问题, 其中  $\mu$  和  $\alpha$  分别是矩阵束  $(A, B)$  的广义特征值和广义特征向量. 从而将式(3)表述的 CCA

求解问题转化为式(6)表达的广义特征值求解问题.

本研究持续求解 CCA 采用如下策略, 随着时间推移, 基于式(6): (1) 递推计算样本协方差分块矩阵  $S_{xy}$ 、 $S_{yx}^{-1}$ 、 $S_{yx}$  和  $S_{xx}$ , 从而得到  $A$  和  $B$  的持续计算式; (2) 对  $B$  进行秩 2 更新, 以并行方式持续计算其特征对; (3) 基于  $B$  的特征对寻找矩阵  $U$ , 使得特征对  $(\mu, \alpha)$  可由  $U$  和  $A$  表达, 从而完成 CCA 的持续求解.

特别地, 典型相关变量对应的另一向量  $\beta$  由式(3)表达, 对应的用样本协方差估计的计算式为:

$$\beta = \frac{1}{\lambda} S_{yy}^{-1} S_{yx} \alpha = \frac{1}{\sqrt{\mu}} S_{yy}^{-1} S_{yx} \alpha \quad (7)$$

显然  $\beta$  也可以通过并行方式持续更新获得. 下面先推导  $A$  和  $B$  的持续计算式, 再研究  $B$  的秩 2 更新, 并寻找满足条件的矩阵  $U$ .

### 4 样本协方差矩阵的递推求解

实际计算中获得的是样本数据, 故求解 CCA 时需使用样本协方差矩阵估计总体协方差矩阵. 由统计知识可知, 前者是后者的无偏估计. 样本数据矩阵如式(5)所示. 记  $l$  时刻样本均值向量和协方差矩阵分别为  $\bar{\mathbf{z}}^{(l)}$  和  $S^{(l)}$ . 先推导样本协方差分块矩阵的递推计算式. 令  $\Delta_z = \mathbf{z}_1^{(l)} - \mathbf{z}_N^{(l-1)}$ ,  $\Delta_x = \mathbf{x}_1^{(l)} - \mathbf{x}_N^{(l-1)}$ ,  $\Delta_y = \mathbf{y}_1^{(l)} - \mathbf{y}_N^{(l-1)}$

其中  $\mathbf{x}_i^{(j)}$  和  $\mathbf{y}_i^{(j)}$  分别为  $\mathbf{z}_i^{(j)}$  的前  $p$  个和后  $q$  个分量. 由于窗口不断前移, 因此相邻窗口间数据关系为  $\mathbf{z}_t^{(l)} = \mathbf{z}_{t-1}^{(l-1)}$ , 显然有  $\bar{\mathbf{z}}^{(l)} = \bar{\mathbf{z}}^{(l-1)} + \frac{1}{N} \Delta_z$ . 此外, 根据样本协方差矩阵定义有  $S^{(l)} = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{z}_i^{(l)} - \bar{\mathbf{z}}^{(l)}) (\mathbf{z}_i^{(l)} - \bar{\mathbf{z}}^{(l)})^T$ , 将右端  $t=1$  的项提到求和符号外并进行适当整理, 且留意到  $\sum_{i=1}^N (\mathbf{z}_i^{(l-1)} - \bar{\mathbf{z}}^{(l-1)}) = 0$ , 则易推得:

$$S^{(l)} = S^{(l-1)} + \Delta_1 \Delta_z^T + \Delta_z \Delta_1^T = S^{(l-1)} + U P U^T \quad (9)$$

其中  $\Delta_1 = \frac{\mathbf{z}_1^{(l)} - \bar{\mathbf{z}}^{(l)}}{N-1} - \frac{\Delta_z}{2N}$ ,  $U = [\Delta_1, \Delta_z]$ ,  $P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ . 上式的分块形式为:

$$S^{(l)} = \begin{bmatrix} S_{xx}^{(l)} & S_{xy}^{(l)} \\ S_{yx}^{(l)} & S_{yy}^{(l)} \end{bmatrix} = \begin{bmatrix} S_{xx}^{(l-1)} + U_x P U_x^T & S_{xy}^{(l-1)} + U_x P U_y^T \\ S_{yx}^{(l-1)} + U_y P U_x^T & S_{yy}^{(l-1)} + U_y P U_y^T \end{bmatrix} \quad (10)$$

其中,  $U_x = [\Delta_{1x}, \Delta_x]$ ,  $U_y = [\Delta_{1y}, \Delta_y]$ ,  $\Delta_{1x} = \frac{\mathbf{x}_1^{(l)} - \bar{\mathbf{x}}^{(l)}}{N-1} - \frac{\Delta_x}{2N}$ ,  $\Delta_{1y} = \frac{\mathbf{y}_1^{(l)} - \bar{\mathbf{y}}^{(l)}}{N-1} - \frac{\Delta_y}{2N}$ .

本文采用式(9)持续更新协方差矩阵. 显然此式是线性递推更新式, 也是精确计算式. 除了求解协方差矩阵的递推更新式外, 还有必要推导样本协方差分块矩

阵的逆矩阵  $(S_{yy}^{(l)})^{-1}$  的递推式. 为此, 先不加证明地介绍矩阵求逆引理. 此引理由 Sherman 和 Morrison 于上世纪中叶推导出, 证明过程可参阅文献[20].

**引理 1** (Sherman-Morrison 公式) 对于  $n \times n$  可逆矩阵  $A$  及  $n \times 1$  向量  $\alpha$  和  $\beta$ , 若  $(A + \alpha\beta^T)$  可逆, 则有

$$(A + \alpha\beta^T)^{-1} = A^{-1} - (1 + \beta^T A^{-1} \alpha)^{-1} A^{-1} \alpha\beta^T A^{-1} \quad (11)$$

**定理 1** (逆矩阵  $(S_{yy}^{(l)})^{-1}$  的递推式) 对于  $q$  阶可逆方阵  $S_{yy}^{(l)} = S_{yy}^{(l-1)} + \Delta_{1y} \Delta_y^T + \Delta_y \Delta_{1y}^T$ , 如果  $S_{yy}^{(l-1)}$  与  $S_{yy}^{(l-1)} + \Delta_{1y} \Delta_y^T$  都可逆, 则  $(S_{yy}^{(l)})^{-1}$  是  $(S_{yy}^{(l-1)})^{-1}$ 、 $y_1^{(l)}$  和  $y_N^{(l-1)}$  的函数, 即

$$\begin{cases} (S_{yy}^{(l)})^{-1} = T^{-1} - (1 + \Delta_{1y}^T T^{-1} \Delta_y)^{-1} T^{-1} \Delta_y \Delta_{1y}^T T^{-1}, \\ T^{-1} = (S_{yy}^{(l-1)})^{-1} - (1 + \Delta_y^T (S_{yy}^{(l-1)})^{-1} \Delta_{1y})^{-1} (S_{yy}^{(l-1)})^{-1} \\ \Delta_{1y} \Delta_y^T (S_{yy}^{(l-1)})^{-1} \end{cases} \quad (12)$$

**证明** 两次利用式(11)可得证. 记  $\hat{A} = S_{yy}^{(l)}$ ,  $A = S_{yy}^{(l-1)}$ ,  $\alpha = \Delta_{1y}$ ,  $\beta = \Delta_y$ ,  $T = A + \alpha\beta^T = S_{yy}^{(l-1)} + \Delta_{1y} \Delta_y^T$ , 有  $\hat{A} = T + \beta\alpha^T$ . 因为  $T = S_{yy}^{(l-1)} + \Delta_{1y} \Delta_y^T$  可逆, 由式(11)得到  $T^{-1} = (A + \alpha\beta^T)^{-1} = A^{-1} - (1 + \beta^T A^{-1} \alpha)^{-1} A^{-1} \alpha\beta^T A^{-1}$ . 同理有  $\hat{A}^{-1} = (T + \beta\alpha^T)^{-1} = T^{-1} - (1 + \alpha^T T^{-1} \beta)^{-1} T^{-1} \beta\alpha^T T^{-1}$ . 带回所设符号即得式(12). 证毕.

定理 1 表明, 第  $l$  时刻逆矩阵  $(S_{yy}^{(l)})^{-1}$  可由前一时刻逆矩阵  $(S_{yy}^{(l-1)})^{-1}$  持续更新而得, 且计算过程无求逆操作 ( $l-1$  时刻的逆  $(S_{yy}^{(l-1)})^{-1}$  为已知量), 只有简单的矩阵乘法和加法运算. 至此, 与样本协方差矩阵相关的递推算式推导完毕, 从而持续求解 CCA 的第一步 (即式(6)对应的矩阵  $A = S_{xy} S_{yy}^{-1} S_{yx}$  和  $B = S_{xx}$  的持续计算) 得以实现. 下面深入研究多维数据流典型相关跟踪算法的若干细节.

## 5 典型相关秩 2 更新算法

本文研究的多维数据流典型相关跟踪算法称为 TCCA (Tracking CCA). TCCA 以秩 2 更新方法持续跟踪多维数据流的相关性. 该方法不仅具有持续更新特性, 而且具有并行求解能力, 这将大幅加快数据流处理速度, 从而提高数据流典型相关跟踪效率. 下文中矩阵  $A$  和  $B$  取值分别为  $S_{xy} S_{yy}^{-1} S_{yx}$  和  $S_{xx}$ , 并假设二者皆为满秩矩阵. 正如研究目的部分所述, CCA 的持续求解主要分三个步骤: (1) 递推更新矩阵  $A$  和  $B$ ; (2) 对  $B$  进行秩 2 更新; (3) 基于  $B$  的特征对寻找矩阵  $U$ , 使得特征对  $(\mu, \alpha)$  可由  $U$  和  $A$  表达. 此三步也即是 TCCA 的关键步骤. 其中第一步在第 4 节已经实现, 下面研究后两步的解决策略, 并给出 TCCA 算法的完整描述.

### 5.1 矩阵 $B$ 的秩 2 更新原理

本节研究  $B$  的特征对的秩 2 更新问题. 所谓特征对的秩 2 更新是指, 当前时刻的特征对是在前一时刻的

特征对基础上, 仅仅以当前时刻流入和流出窗口的数据为参数更新而得.  $B$  的特征对将以持续更新和并行方式求解, 这是加快数据流处理速度的有效途径.

记  $\hat{B} = S_{xx}^{(l)}$ ,  $B = S_{xx}^{(l-1)}$ , 由式(10)有  $\hat{B} = B + \Delta_x \Delta_{1x}^T + \Delta_{1x} \Delta_x^T$ . 设  $B$  的特征分解式为  $B = V D V^T$ ,  $\alpha = V^T \Delta_x$ ,  $\beta = V^T \Delta_{1x}$ ,  $T = V^T \hat{B} V = D + \alpha\beta^T + \beta\alpha^T$ , 则  $T$  的特征多项式为  $\varphi(\lambda) = \det(T - \lambda I) = \det(D - \lambda I) \psi(\lambda)$ , 其中  $\psi(\lambda) = \det(I + (D - \lambda I)^{-1}(\alpha\beta^T + \beta\alpha^T))$ , 整理得:

$$\psi(\lambda) = 1 - 2 \sum_{i=1}^p \frac{\alpha_i \beta_i}{\lambda - d_i} - \sum_{i=1}^p \sum_{j=i+1}^p \frac{(\alpha_i \beta_j - \alpha_j \beta_i)^2}{(\lambda - d_i)(\lambda - d_j)} \quad (13)$$

式中,  $\alpha_i$  和  $\beta_i$  分别是  $\alpha$  和  $\beta$  的第  $i$  个分量, 而  $d_i$  是更新前矩阵  $B$  的第  $i$  个特征值. 由上式求解关于  $\lambda$  的方程  $\psi(\lambda) = 0$ , 即得更新后  $S_{xx}^{(l)}$  对应的新特征值. 显然式(13)与式(4)差异明显, 前者是基于协方差精确更新的计算式, 而后者是基于协方差估计的计算式.

一个很具有吸引力的事实是, 可通过并行方式求解关于  $\lambda$  的方程  $\psi(\lambda) = 0$ , 即  $S_{xx}^{(l)}$  的所有特征值  $\lambda_i$  能以并行方式快速解出. 为此, 先不加证明地介绍一个重要的定理, 即广义交织定理 (Generalized Interleaf Theorem).

**定理 2** (广义交织定理)<sup>[21]</sup> 对  $p \times p$  矩阵  $E = ww^T + vv^T$  和对称矩阵  $G$ , 有  $\lambda_{k-1}(G) \leq \lambda_k(G + E) \leq \lambda_{k+1}(G)$ , 其中  $k = 1, 2, \dots, p$ . 记  $\eta = \max(\|w\|^2, \|v\|^2)$ , 则  $\lambda_0(G) = \lambda_1(G) - \eta$ ,  $\lambda_{p+1}(G) = \lambda_p(G) + \eta$ .

广义交织定理表明, 更新后矩阵的每个特征值分别介于更新前矩阵的两个不同特征值包含的区间内, 而且这些区间除边界外没有其它交集, 因此可通过搜索方式在更新前特征值包含的区间内寻找更新后的特征值; 此外, 由于旧特征值所包含区间的无交连性, 每个新特征值可以在并行环境下不同计算节点中并行解出.

求解新特征值的具体步骤是, 先根据广义交织定理, 将旧特征值  $d_i (i = 1, 2, \dots, p)$  分割为互不交连的区间  $[d_i, d_{i+1}] (i = 0, 1, \dots, p-1)$ , 其边界值将作为求解方程  $\psi(\lambda) = 0$  的搜索边界; 再将不同区间边界值和式(13)分别送入并行环境下的不同计算节点, 并行地用 Newton 法在对应区间内搜索新特征值. Newton 法的二次收敛性可保证求解快速完成, 常常只需一次搜索即可找到满足精度要求的解, 这对于数据流快速处理要求是适宜的.

而关于特征向量的更新, 二步法具有通用性<sup>[19]</sup>. 若  $\hat{B}$  的特征方程为  $(\hat{B} - \lambda_i I) x_i = 0$ , 则有  $(B - \lambda_i I) x_i + U_x P U_x^T x_i = 0$ , 其中  $U_x$  和  $P$  同式(10). 引入二维列向量  $y_i$ , 从而有方程  $(B - \lambda_i I) x_i + U_x y_i = 0$  和  $U_x^T x_i - P^{-1} y_i = 0$ . 代入  $B$  的特征分解式  $B = V D V^T$  可得:

$$\mathbf{x}_i = -\mathbf{V}(\mathbf{D} - \lambda_i \mathbf{I})^{-1} \mathbf{V}^T \mathbf{U}_x \mathbf{y}_i \quad (14)$$

上式应用了  $\mathbf{V}$  的正交性质,即  $\mathbf{V}^{-1} = \mathbf{V}^T$ . 又因为  $\mathbf{P}^{-1} = \mathbf{P}$ , 所以有

$$(\mathbf{P} + \mathbf{U}_x^T \mathbf{V}(\mathbf{D} - \lambda_i \mathbf{I})^{-1} \mathbf{V}^T \mathbf{U}_x) \mathbf{y}_i = 0 \quad (15)$$

显然,式(15)是一个二元方程组,求解  $\mathbf{y}_i$  是极其容易且快速的. 所谓二步法是指,基于已经计算出的  $\hat{\mathbf{B}}$  的特征值  $\lambda_i$ ,分两步求解  $\lambda_i$  对应的特征向量  $\mathbf{x}_i$ . 首先由式(15)求出  $\mathbf{y}_i$ ,其次将  $\mathbf{y}_i$  代入式(14)即得  $\mathbf{x}_i$ ,再将  $\mathbf{x}_i$  归一化即得更新后的特征向量  $\hat{\mathbf{v}}_i = \mathbf{x}_i / \|\mathbf{x}_i\|_2$ .

一个令人欣慰的事实是,  $\hat{\mathbf{B}}$  的不同特征值对应的特征向量也可通过并行方式解出. 观察式(15),由于  $\lambda_i$  已通过并行方式解出,是已知量,因此可将此方程送入并行环境下不同计算节点,以并行方式快速解出  $\mathbf{y}_i$ . 而  $\mathbf{x}_i$  以及归一化后的特征向量  $\hat{\mathbf{v}}_i$  只需在解出  $\mathbf{y}_i$  的当前节点中求解即可,显然不同的  $\hat{\mathbf{v}}_i$  也是并行解出的.

至此,我们得出的结论是,矩阵  $\mathbf{B}$  的特征分解可通过并行方式快速持续更新,这恰是数据流快速处理要求所欢迎的,也是本文引入秩 2 更新理论的初衷所在. 下面以算法形式描述矩阵  $\mathbf{B}$  的特征对的更新步骤.

**算法 1** 矩阵  $\mathbf{B}$  的秩 2 更新算法:

Input:  $l-1$  时刻  $\mathbf{B} = \mathbf{S}_{xx}^{(l-1)}$  的特征分解矩阵  $\mathbf{V}$  和  $\mathbf{D} = \text{diag}(d_1, d_2, \dots, d_p)$  (其中  $\mathbf{V}^T \mathbf{B} \mathbf{V} = \mathbf{D}$  为  $\mathbf{B}$  的特征分解式),以及  $\mathbf{A}_x$  和  $\mathbf{A}_{1x}$  (取值见式(10)).

Output:  $l$  时刻  $\mathbf{S}_{xx}^{(l)}$  的特征分解矩阵  $\mathbf{V}$  和  $\mathbf{D}$ .

Step 1. 置  $\boldsymbol{\alpha} = \mathbf{V}^T \mathbf{A}_x$ ,  $\boldsymbol{\beta} = \mathbf{V}^T \mathbf{A}_{1x}$ ;

Step 2. 并行求解 (如 Newton 法) 关于  $\lambda$  的方程 1 -

$$2 \sum_{i=1}^p \frac{\alpha_i \beta_i}{\lambda - d_i} - \sum_{i=1}^p \sum_{j=i+1}^p \frac{(\alpha_i \beta_j - \alpha_j \beta_i)^2}{(\lambda - d_i)(\lambda - d_j)} = 0;$$

Step 3. 根据不同  $\lambda_i$  值,以并行方式解出式(15)对应的不同  $\mathbf{y}_i$  值,代入式(14)解出  $\mathbf{x}_i$ ,并进行归一化得更新后的特征向量  $\hat{\mathbf{v}}_i (i = 1, 2, \dots, p)$ ;

Step 4. 置  $\mathbf{D}$  中  $d_i$  为新值  $\lambda_i$ ,置  $\mathbf{V}$  中列向量  $\mathbf{v}_i$  为  $\hat{\mathbf{v}}_i (i = 1, 2, \dots, p)$ . 算法结束.

此算法中,第二步并行求解方程时,Newton 法的二次收敛性只需有限次 (常常只需一次) 搜索即可找到满足精度要求的解,从而算法复杂度主要由第 3 步确定. 式(15)的求逆对象是对角阵,只需  $p$  次运算. 因此,算法 1 的时间和空间复杂度显然都为  $O(p^2)$ .

## 5.2 矩阵 $\mathbf{U}$ 的求解算法

本节基于  $\mathbf{B}$  的特征对,寻找用于表达式(6)对应特征对的矩阵  $\mathbf{U}$ ,  $\mathbf{U}$  和  $\mathbf{A}$  将一起用于表达矩阵束  $(\mathbf{A}, \mathbf{B})$  的特征对. 正如研究目的部分所述,此特征对的求解是 CCA 求解的转化形式. TCCA 的跟踪对象,也即两数据流之间的典型相关系数和对应的投影向量,将由此特征对直接导出. 因此矩阵  $\mathbf{U}$  的求解也是 TCCA 算法的

重要步骤之一. 下面先介绍几个概念和定理.

**定义 1** (正则矩阵束) 将  $p \times p$  的方阵  $\mathbf{A}$  和  $\mathbf{B}$  构成的二元组  $(\mathbf{A}, \mathbf{B})$  称为矩阵束 (Matrix Pencil). 若  $\mathbf{A}$  和  $\mathbf{B}$  均为 Hermitian 矩阵,且  $\mathbf{B}$  正定 (即  $\mathbf{B} > 0$ ),则称  $(\mathbf{A}, \mathbf{B})$  为正则矩阵束 (Regular Matrix Pencil).

**定义 2** (广义特征值) 给定矩阵束  $(\mathbf{A}, \mathbf{B})$ ,对于标量  $\lambda$ ,若存在非零向量  $\mathbf{u}$ ,使得  $\mathbf{A}\mathbf{u} = \lambda \mathbf{B}\mathbf{u}$ ,则称标量  $\lambda$  和非零向量  $\mathbf{u}$  分别为矩阵束  $(\mathbf{A}, \mathbf{B})$  的广义特征值和广义特征向量.

**定理 3**<sup>[21]</sup> 对于正则矩阵束  $(\mathbf{A}, \mathbf{B})$ ,存在矩阵  $\mathbf{U}$ ,使得  $\mathbf{U}^H \mathbf{B} \mathbf{U} = \mathbf{I}_p$ ,  $\mathbf{U}^H \mathbf{A} \mathbf{U} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$ . 其中,  $\lambda_1, \lambda_2, \dots, \lambda_p$  为  $(\mathbf{A}, \mathbf{B})$  的广义特征值,而  $\mathbf{U}^H$  为  $\mathbf{U}$  的复共轭转置.

**定理 4**<sup>[22]</sup> 设容量为  $n$  的  $p$  元正态总体  $\mathbf{X} \sim N_p(\mathbf{0}, \mathbf{C})$  的样本协方差矩阵为  $\mathbf{B}$ ,则  $P\{\mathbf{B} > 0\} = 1 \Leftrightarrow n > p$ .

定理 3 和定理 4 的证明过程可参阅文献[21, 22]. 一般地,样本容量  $n$  小于或等于维数  $p$  的情况并不多见. 因此定理 4 中的条件  $n > p$  通常能得到保证. 当  $\mathbf{A}$  和  $\mathbf{B}$  取值分别为  $\mathbf{S}_{xy} \mathbf{S}_{yy}^{-1} \mathbf{S}_{yx}$  和  $\mathbf{S}_{xx}$  时均为 Hermitian 矩阵,且  $\mathbf{B}$  依概率为正定矩阵 (即  $P\{\mathbf{B} > 0\} = 1$ ),故式(6)中用样本协方差分块矩阵表达的矩阵束  $(\mathbf{A}, \mathbf{B})$  在概率意义上能保证其正则性.

定理 3 中,若记  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p]$ ,则有  $\mathbf{A}\mathbf{u}_i = \lambda_i \mathbf{B}\mathbf{u}_i$ ,所以  $\mathbf{u}_i$  即为正则矩阵束  $(\mathbf{A}, \mathbf{B})$  关于广义特征值  $\lambda_i$  的广义特征向量. 因此寻找矩阵  $\mathbf{U}$  即可解出正则矩阵束  $(\mathbf{A}, \mathbf{B})$  的广义特征值和广义特征向量,从而由列向量  $\mathbf{u}_i$  构成的矩阵  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p]$  即为所寻找的可用于表达式(6)对应特征对的矩阵.

关于满足定理 3 条件的矩阵  $\mathbf{U}$  的寻找,文献[21]中给出求解算法,其步骤为:(1)对  $\mathbf{B}$  进行 Cholesky 分解  $\mathbf{B} = \mathbf{G}\mathbf{G}^T$ ;(2)计算  $\mathbf{T} = \mathbf{G}^{-1} \mathbf{A} \mathbf{G}^{-T}$ ;(3)对  $\mathbf{T}$  进行 Schur 分解  $\mathbf{Q}^T \mathbf{T} \mathbf{Q} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$ ;(4)置  $\mathbf{U} = \mathbf{G}^{-1} \mathbf{Q}$ . 但此算法中第一步对  $\mathbf{B}$  进行 Cholesky 分解  $\mathbf{B} = \mathbf{G}\mathbf{G}^T$  无法通过持续更新的方式进行,幸运的是,文献[23]分析了用  $\mathbf{V}\mathbf{D}^{-1/2}$  代替  $\mathbf{G}$  的合理性,其中  $\mathbf{V}^T \mathbf{B} \mathbf{V} = \mathbf{D}$  是  $\mathbf{B}$  的特征分解式. 因为特征分解可通过秩 2 更新的方式持续求解,故本文用  $\mathbf{V}\mathbf{D}^{-1/2}$  代替  $\mathbf{G}$ ,对文献[21]中的前述算法进行改进,使其能持续计算矩阵  $\mathbf{U}$ . 特别地,由于  $\mathbf{D}$  为对角矩阵,故有  $\mathbf{D}^{1/2} = \text{diag}(\sqrt{d_1}, \sqrt{d_2}, \dots, \sqrt{d_p})$ ;另一方面,  $\mathbf{V}$  是正交矩阵,从而有  $\mathbf{V}^{-1} = \mathbf{V}^T$ ,所以  $\mathbf{G}^{-1} = (\mathbf{V}\mathbf{D}^{-1/2})^{-1} = \mathbf{D}^{1/2} \mathbf{V}^{-1} = \text{diag}(\sqrt{d_1}, \sqrt{d_2}, \dots, \sqrt{d_p}) \mathbf{V}^T$ . 下面给出求解矩阵  $\mathbf{U}$  的算法步骤.

**算法 2** 矩阵  $\mathbf{U}$  的求解算法:

Input:  $p \times p$  阶矩阵  $\mathbf{A}$ 、 $\mathbf{V}$  和  $\mathbf{D}$  (其中  $\mathbf{V}^T \mathbf{B} \mathbf{V} = \mathbf{D} =$

$\text{diag}(\sqrt{d_1}, \sqrt{d_2}, \dots, \sqrt{d_p})$  是  $\mathbf{B}$  的特征分解).

Output: 正则矩阵束  $(\mathbf{A}, \mathbf{B})$  的广义特征向量矩阵  $\mathbf{U}$  和广义特征值  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$ .

Step 1. 置  $\mathbf{G}_{\text{inv}} = \text{diag}(\sqrt{d_1}, \sqrt{d_2}, \dots, \sqrt{d_p}) \mathbf{V}^T$ ;

Step 2. 置  $\mathbf{T} = \mathbf{G}_{\text{inv}} \mathbf{A} \mathbf{G}_{\text{inv}}^T$ ;

Step 3. 对  $\mathbf{T}$  进行 Schur 分解  $\mathbf{Q}^T \mathbf{T} \mathbf{Q} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p) = \mathbf{\Lambda}$ ;

Step 4. 置  $\mathbf{U} = \mathbf{G}_{\text{inv}} \mathbf{Q}$ . 算法结束.

此算法的执行时间主要花费在第三步. 若用 Hessenberg 约化方法求解<sup>[21]</sup>, 则 Schur 分解每次迭代时间复杂度可由  $O(p^3)$  降低为  $O(p^2)$ , 最坏情况下迭代不超过  $p$  次. 因此算法 2 的时间复杂度为  $O(p^3)$ , 而空间复杂度显然为  $O(p^2)$ .

### 5.3 算法 TCCA

本节给出 TCCA 算法的完整描述. TCCA 以持续更新方式跟踪数据流的相关性, 该算法基于滑动窗口, 根据前一时刻的相关状态和当前时刻流入和流出窗口的数据, 求解当前时刻的相关状态. 此算法具有维持状态的能力, 并且在算法的关键步骤中采用秩 2 更新方法, 以持续、并行方式求解样本协方差矩阵的特征对.

算法 3 TCCA 算法:

Input:  $l-1$  时刻协方差矩阵  $\mathbf{S}^{(l-1)}$ 、分块逆矩阵  $(\mathbf{S}_{yy}^{(l-1)})^{-1}$ 、 $\mathbf{S}_{xx}^{(l-1)}$  的特征分解矩阵  $\mathbf{V}$  和  $\mathbf{D}$ 、窗口中删除的数据  $\mathbf{z}_N^{(l-1)}$  和插入的数据  $\mathbf{z}_1^{(l)}$ .

Output:  $l$  时刻数据流  $\mathbf{X}^{(l)}$  和  $\mathbf{Y}^{(l)}$  之间的典型相关系数  $\rho_k$  和对应的投影向量  $\mathbf{\alpha}_k$  和  $\mathbf{\beta}_k$ , ( $k=1, 2, \dots, p$ ), 并输出  $l$  时刻的协方差矩阵  $\mathbf{S}^{(l)}$ 、分块逆矩阵  $(\mathbf{S}_{yy}^{(l)})^{-1}$  以及  $\mathbf{S}_{xx}^{(l)}$  的特征分解矩阵  $\mathbf{V}$  和  $\mathbf{D}$ .

Step 1. 按式(8)计算  $\mathbf{\Delta}_z$  和  $\mathbf{\Delta}_1$  (注:  $\mathbf{\Delta}_1$  的表达式见式(9));

Step 2. 由式(9)递推求解  $l$  时刻的矩阵  $\mathbf{S}^{(l)}$ ;

Step 3. 由式(12)递推解出  $(\mathbf{S}_{yy}^{(l)})^{-1}$ ;

Step 4. 置  $\mathbf{A} = \mathbf{S}_{xy}^{(l)} (\mathbf{S}_{yy}^{(l)})^{-1} \mathbf{S}_{yx}^{(l)}$  和  $\mathbf{B} = \mathbf{S}_{xx}^{(l)}$ ;

Step 5. 调用算法 1, 更新矩阵  $\mathbf{V}$  和  $\mathbf{D}$ ;

Step 6. 调用算法 2, 求解矩阵束  $(\mathbf{A}, \mathbf{B})$  特征值  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$  及特征向量  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p]$ ;

Step 7. 置  $\mathbf{\alpha}_k = \mathbf{u}_k$ ,  $\mathbf{\beta}_k = \frac{1}{\rho_k} (\mathbf{S}_{yy}^{(l)})^{-1} \mathbf{S}_{yx}^{(l)} \mathbf{\alpha}_k$ ,  $\rho_k = \sqrt{\lambda_k}$ , ( $k=1, 2, \dots, p$ ). 算法结束.

观察算法 TCCA 的输入和输出不难发现, TCCA 维持的状态主要包括四部分: 协方差矩阵  $\mathbf{S}$ 、分块矩阵  $\mathbf{S}_{yy}$  的逆矩阵  $\mathbf{S}_{yy}^{-1}$  以及  $\mathbf{S}_{yy}$  的特征分解矩阵  $\mathbf{V}$  和  $\mathbf{D}$ . 其中  $\mathbf{S}$  于第 2 步以简单的线性操作方式更新;  $\mathbf{S}_{yy}^{-1}$  于第 3 步以并不涉及求逆操作的方式更新; 而  $\mathbf{V}$  和  $\mathbf{D}$  于第 5 步以秩 2 更新方式并行求解. 此外, 最后两步求解 TCCA 在

当前窗口内的跟踪结果, 即当前时刻两数据流之间的典型相关系数和对应的投影向量. 特别地, 第 5 步解出的  $\mathbf{V}$  和  $\mathbf{D}$  作为第 6 步重要的输入参数, 前一步的持续求解和并行性保证了后一步求解的快速性.

算法 3 中, 前两步仅涉及向量乘法和矩阵加法, 其时间和空间复杂度皆为  $O(p+q)$ ; 第三步看似复杂, 其实也仅涉及矩阵乘法和加法, 其时间和空间复杂度分别为  $O(p^3)$  和  $O(p^2)$ ; 第四步和第七步则为矩阵乘法, 其时间和空间复杂度分别为  $O(pq^2)$  和  $O(q^2)$ , (注:  $p \leq q$ ); 而其余两步的复杂度已在前面分析. 因此, TCCA 算法的时间复杂度为  $O(pq^2)$ , 而空间复杂度为  $O(q^2)$ .

需要特别说明的是,  $p$  和  $q$  是维数, 多维数据流的维数多数情况下远远小于数据规模  $N$  (即滑动窗口长度). 数据流处理对算法的时空复杂性一般达到  $\log(N)$  级别即比较理想. 然而, 前段分析结果表明, TCCA 算法的复杂度皆与数据规模  $N$  无关, 而仅与数据流维数有关, TCCA 的效率可见一斑. TCCA 高效的原因在于, 算法的每一步皆未涉及高复杂度运算, 多数以持续更新方式完成, 且关键步骤采用并行方式求解. 每到达一条新数据, 即调用一次 TCCA 算法. 调用 TCCA 过程中, 上一次的状态将作为下一次的输入, 下一次的状态只涉及上一次的状态和插入及删除的数据, 其更新是快速的, 并不需要基于滑动窗口内的所有原始数据重新计算. 显然 TCCA 的高效性能满足数据流快速处理的要求.

特别地, 在启动 TCCA 算法时 (即第一次调用),  $\mathbf{S}_{xx}^{(0)}$  的特征分解矩阵  $\mathbf{V}$  和  $\mathbf{D}$ 、 $\mathbf{S}^{(0)}$  以及  $(\mathbf{S}_{yy}^{(0)})^{-1}$  需要基于第一个窗口内的所有原始数据直接计算. 因此第一个窗口内的执行时间会明显偏高, 但这并不会对数据流处理带来太大影响, 因为执行时间偏高仅此一次, 后续调用过程中将以更新方式快速完成.

## 6 实验及结果分析

本节通过仿真实验对 TCCA 算法进行分析, 重点分析算法的稳定性、执行时间以及在不同参数下的计算效率和精度. 将未采用更新方式实现的 CCA 称为 Standard CCA. 实验将本文提出的算法 TCCA 分别与 Standard CCA 和 ApproxCCA<sup>[15]</sup> 进行对比分析. 实验计算机配置为双核 2.66GHz CPU、2.0GB 内存, 操作系统为 Windows 7 Professional, 软件实现工具为 Matlab R2011a.

实验采用如下三种数据集: ① 高斯数据集 DSg (Gaussian Data Sets). 通过随机方式产生的满足高斯分布的人工数据集, 其中  $X$  和  $Y$  的每一维满足  $N(0, 1)$  分布, 期望相关系数值在 0.5 附近, 而维数和容量将根据实验目的不同而有所差异; ② 带有噪声的线性数据集 DSn (Noisy Linear Data Sets). 也是通过随机方式产生的

人工数据集,在线性数据上添加满足  $N(5,10)$  分布的噪声数据,期望相关系数值  $>0.5$ ,维数和容量根据不同实验目的而有所差别;③臭氧含量检测数据集 DSo(Ozone Level Detection Data Set). 这是一个 UCI 标准数据集 (<http://archive.ics.uci.edu/ml/machine-learning-databases/ozone/>),该数据集包括两组数据,分别记录了 1 小时和 8 小时的观测值,样本容量为 2536,而维数为 74,期望相关系数值接近 1.

由于在时间推移过程中,典型相关系数的变化规律体现了多维数据流相关性的演化轨迹,特别是第 1 典型相关系数更是如此,而典型相关变量在描述数据流相关性变化规律方面则没有直观意义,因此实验部分仅关注第 1 典型相关系数的变化规律,而对于典型相关变量以及其它典型相关系数的变化规律则不予以考虑.

6.1 算法稳定性分析

所谓算法稳定性,在本文中特指算法在每次执行过程中所花费时间的差异性. 标准差是度量差异性的常用指标,本实验用多次重复实验平均运行时间的标准差作为稳定性的度量值. 由于需要通过多次重复的方式对平均运行时间进行分析,故采用第一种数据集,即高斯数据集 DSg. 本实验生成的是低维数据,样本容量为适度容量,具体地,  $X$  和  $Y$  的维数分别为 5 维和 8 维,数据流总长度为 1100,而窗口宽度设置为 1000. 实验通过一次产生数据和重复 20 次产生数据两种方式进行对比分析.

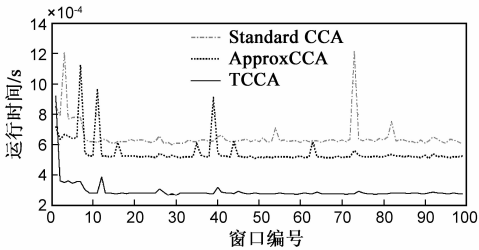


图1 不同窗口下执行时间

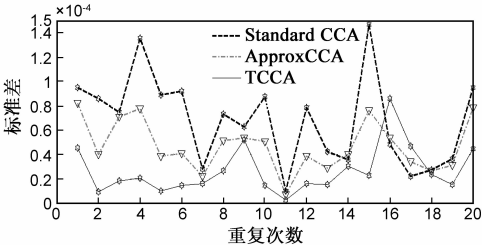


图2 重复运行20次执行时间标准差

图 1 是在一次产生数据方式下相邻窗口内执行时间连线图,共 100 个窗口. 此图至少呈现出两个事实,其一,除第一个窗口外,其余窗口内 TCCA 的执行时间皆显著少于 Standard CCA 和 ApproxCCA 的执行时间,而

且,尽管在第一个窗口内执行时间后两者比 TCCA 略少,但优势并不显著. 其二, TCCA 的执行时间波动较小,在每个窗口内所花费的时间几乎一致,而后两者在不同窗口内的执行时间波动较大,图中, ApproxCCA 至少在 7 个窗口内波动较大,而且其中 3 个窗口内所花费时间接近临近窗口内所花费时间的两倍;而 Standard CCA 也至少有 4 次较大波动. 分析执行时间波动性差异的原因,其根源在于 Standard CCA 和 ApproxCCA 在每个窗口内都有一次矩阵求逆操作,而求逆操作是不稳定的,但 TCCA 除第一个窗口外,并不要求求逆,其分块逆矩阵的计算在前一次基础上通过持续更新方式完成,是稳定的.

而图 2 是在重复 20 次产生数据方式下,每次运行时间标准差对比图,每次运行的数据重新随机生成. 由图不难看出,除第 16 次和第 17 次外,其余各次运行过程中 Standard CCA 和 ApproxCCA 执行时间标准差几乎都显著大于 TCCA 花费时间标准差,说明前两者没有 TCCA 稳定,同时也表明图 1 中所呈现出的波动性差异并非偶然. 此外还发现, TCCA 执行时间标准差几乎在  $6 \times 10^{-5}$  以下,且多数小于  $2 \times 10^{-5}$ ,这是一个较小的数值,表明 TCCA 算法的稳定性较好.

6.2 算法执行时间对比

本实验将三种算法(即 TCCA、ApproxCCA 和 Standard CCA)在前述三种数据集(即 DSg、DSn 和 DSo)上对每个窗口内平均执行时间进行对比分析. 对于数据集 DSg 和 DSn,  $X$  和  $Y$  的维数以及数据流总长度皆与数据集 DSo 相同,即维数为 71,数据流总长度为 2536. 而窗口宽度统一设置为 1000.

每种数据集上平均执行时间如图 3 所示. 显然,每种数据集上, TCCA 在每个窗口内的平均处理时间皆小于其它两种算法的处理时间. 此现象的成因是显然的,因为 TCCA 算法通过更新方式实现,其协方差矩阵以及分块逆矩阵等的计算都在原有基础上通过更新方式完成,计算量较小,并不像另外两种算法那样基于窗口内的所有数据重新计算,计算开销较大. 因此可以认为 TCCA 的执行速度比 ApproxCCA 和 Standard CCA 的快.

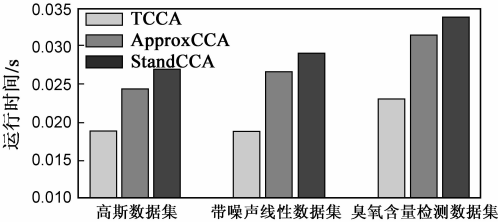


图3 三种数据集上平均执行时间

6.3 窗口长度对精度和运行时间的影响

本实验在不同窗口长度下,分别在三种数据集上(即 DSg、DSn 和 DSo)对三种算法(即 TCCA、ApproxCCA

和 Standard CCA) 计算精度和运行时间进行对比分析. 对于 DSg 和 DSn 数据集,  $X$  和  $Y$  的维数以及数据流总长度皆设置为与数据集 DSo 相同. 窗口宽度初值设置为 1000, 步长为 300, 最大值为 2500.

需要说明的是, 尽管 TCCA 也是精确算法, 但由于受计算机字长限制, TCCA 和 Standard CCA 计算结果与真实值之间皆存在偏差. 本实验以及下一实验比较精度时, 皆以 Standard CCA 为参照标准, 其中所谓的“精度”并不是 TCCA 以及 ApproxCCA 计算结果与真实值之

间的偏差, 而是与 Standard CCA 计算结果的差异.

计算精度在不同窗口长度下的变化趋势如图 4 所示. 由此图可以看出, 随着窗口长度增加, TCCA 和 ApproxCCA 在每种数据集上的精度都在缓慢提高 (即误差下降). 窗口长度值即是样本容量, 窗口长度增加意味着样本容量加大, 而样本容量加大, 误差降低, 这与统计学的一般规律是相符的. 因此, 适度增加窗口长度有利于降低计算误差. 此外, 与 ApproxCCA 相比, TCCA 的计算精度有明显优势.

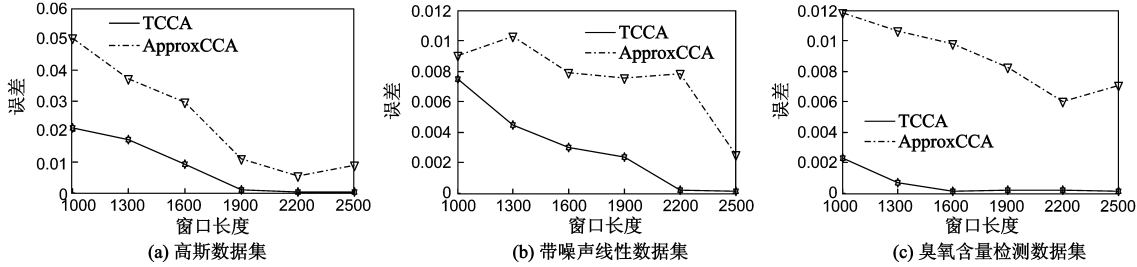


图4 不同窗口长度下计算精度

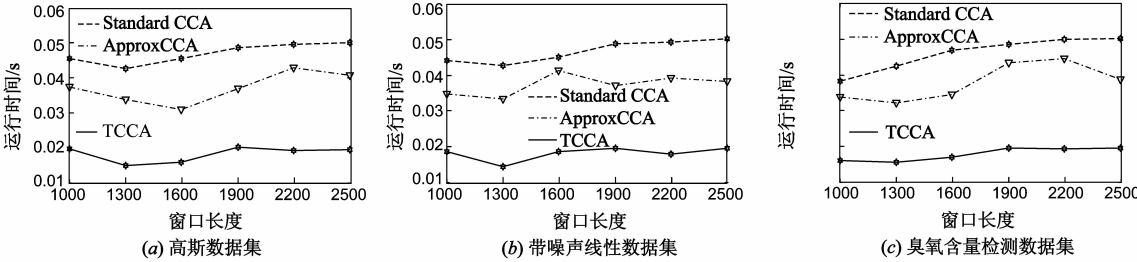


图5 不同窗口长度下平均执行时间

而不同长度窗口下的运行时间对比如图 5 所示. 由图可知, 一方面, 与其它两种算法相比, 不同数据集下, TCCA 的平均执行时间都显著较少; 另一方面, Standard CCA 和 ApproxCCA 平均执行时间随着窗口长度增加而上升的趋势相对明显, 而 TCCA 平均执行时间的上升幅度却非常有限, 说明 TCCA 的时间性能对于一定范围内窗口长度的敏感性较小, 这为窗口长度的选择放宽了条件.

6.4 维数对精度和运行时间的影响

本实验分别在 DSg、DSn 和 DSo 三种数据集上, 分析维数对三种算法 (即 TCCA、ApproxCCA 和 Standard CCA) 计算精度和运行时间的影响. 数据流总长度为 2536, 窗口宽度设置为 1000. 实验在 10 种不同的维数下进行比较分析, 维数通过随机方式生成, 生成规则为: 先产生两组容量为 10 的随机整数  $r_1$  和  $r_2$ , 范围介于 1 至 71 之间; 第  $i$  ( $i = 1, 2, \dots, 10$ ) 次执行时, 将  $r_1$  和  $r_2$  中第  $i$  个值中的小者和大者分别赋予  $X$  的维数  $P_x$  和  $Y$  的维数  $P_y$ . 对于数据集 DSg 和 DSn, 每次重新生成; 而对于数据集 DSo, 每次分别从 1 小时和 8 小时数据集中随机抽取  $P_x$  列和  $P_y$  列. 表 1 是实验中随机生成的关于  $X$  和  $Y$  的

维数  $P_x$  和  $P_y$  的情况.

表 1 随机生成的  $X$  和  $Y$  的维数

| 序号          | d1 | d2 | d3 | d4 | d5 | d6 | d7 | d8 | d9 | d10 |
|-------------|----|----|----|----|----|----|----|----|----|-----|
| $P_x$       | 10 | 8  | 13 | 16 | 29 | 21 | 15 | 24 | 33 | 57  |
| $P_y$       | 12 | 19 | 15 | 44 | 39 | 48 | 55 | 57 | 60 | 61  |
| $P_x + P_y$ | 22 | 27 | 28 | 60 | 68 | 69 | 70 | 81 | 93 | 118 |

总维数对计算精度的影响如图 6 所示. 就总体趋势而言, 一则误差并不显著, 差值几乎在 0.01 以下; 二则随着总维数的增加, 误差略有增大, 但并非呈线性增加趋势; 三则 ApproxCCA 的误差显然大于 TCCA 的误差. 总之, 可以认为, 增大总维数, 误差加大的风险也升高, 但升高的幅度并不显著.

而不同维数下的运行时间对比如图 7 所示. 由图可以看出, 一方面, 不同数据集下 TCCA 的平均执行时间都显著小于 Standard CCA 和 ApproxCCA 的执行时间; 另一方面, Standard CCA 和 ApproxCCA 平均执行时间随着窗口长度增加而上升的趋势是显著的, 而 TCCA 平均执行时间的上升趋势却非常缓慢, 在三种数据集下都只有在最后一个维数 (即总维数达 118) 才显露出升高的趋势, 说明 TCCA 的时间性能对于一定范围内维数的敏感性也较小. 此外, 比较图 5 和图 7 不难发现, Standard

CCA 和 ApproxCCA 的执行时间对维数比对窗口长度更敏感,而 TCCA 并无此缺陷。

总之,仿真实验结果表明,TCCA 具有较好的稳定性、较高的计算效率和精度,该算法是有效、可行的。

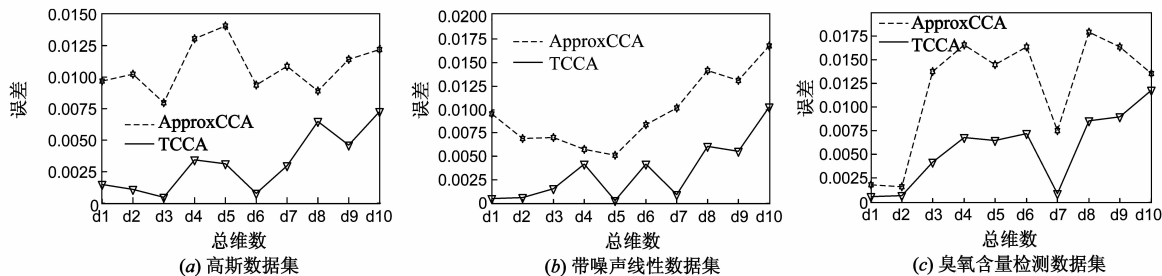


图6 不同维数下计算精度

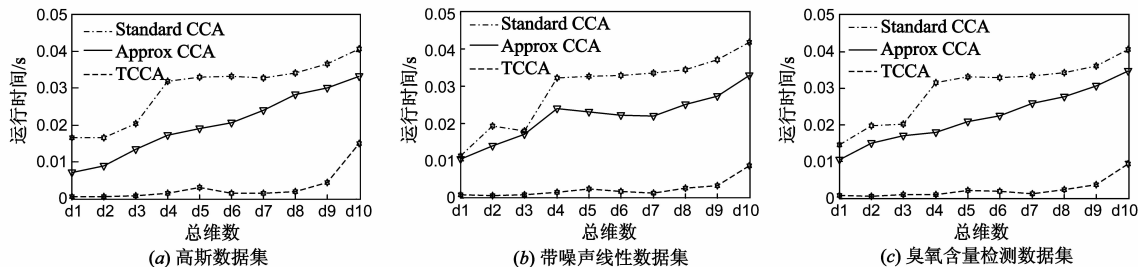


图7 不同维数下平均执行时间

## 7 结论及展望

本文基于秩 2 更新理论,提出了跟踪多维数据流典型相关的算法 TCCA。TCCA 通过并行方式持续更新两组多维数据流中维数较低的数据流窗口对应的协方差矩阵的特征子空间,在此基础上对多维数据流的典型相关系数和典型相关变量进行持续跟踪。仿真实验结果表明,TCCA 具有较好的稳定性和较高的计算效率。

但本文研究的算法未涉及滞后相关性的跟踪,多维数据流的滞后相关性和单维数据流的滞后相关性一样,也是现实应用中常见的关系之一,因此多维数据流的滞后相关性跟踪将是我们下一步的研究课题。

### 参考文献

- [1] Sakurai Y, Papadimitriou S, et al. BRAID: stream mining through group lag correlations [A]. Jennifer W. SIGMOD 2005: Proceedings of the ACM SIGMOD International Conference on Management of Data [C]. New York: ACM Press, 2005. 599 – 610.
- [2] Sakurai Y, Faloutsos C, et al. Fast discovery of group lag correlations in streams [J]. ACM Transactions on Knowledge Discovery from Data, 2010, 5(1): Article 5.
- [3] Yunyue Zhu, Dennis Shasha. StatStream: statistical monitoring of thousands of data streams in real time [A]. Philip A B. VLDB 2002: Proceedings of the 28th International Conference on Very Large Data Bases [C]. Hong Kong SAR: Elsevier Press, 2002. 358 – 369.
- [4] Papadimitriou S, Jimeng Sun, et al. Streaming pattern discovery in multiple time-series [A]. Klemens B. VLDB 2005: Proceed-

ings of 31st International Conference on Very Large Data Bases [C]. New York: ACM Press, 2005. 697 – 708.

- [5] Tiancheng Zhang, Dejun Yue, et al. Adaptive correlation analysis in stream time series with sliding windows [J]. Computers and Mathematics with Applications, 2009, 57(6): 937 – 948.
- [6] Mueen A, Nath S, et al. Fast approximate correlation for massive time-series data [A]. Ahmed E. SIGMOD 2010: Proceedings of the 2010 International Conference on Management of Data [C]. New York: ACM Press, 2010. 171 – 182.
- [7] Galen Reeves, Jie Liu, et al. Managing massive time series streams with multi-scale compressed trickles [J]. Proceedings of the VLDB Endowment, 2009, 2(1): 97 – 108.
- [8] Harold Hotelling. Relations between two sets of variates [J]. Biometrika, 1936, 28(3): 321 – 377.
- [9] 孙权森, 曾生根, 等. 典型相关分析的理论及其在特征融合中的应用 [J]. 计算机学报, 2005, 28(9): 1524 – 1533.
- [10] SUN Quran-Sen, ZENG Sheng-Gen, et al. The theory of canonical correlation analysis and its application to feature fusion [J]. Chinese Journal of Computers, 2005, 28(9): 1524 – 1533. (in Chinese)
- [11] Chaudhuri K, Kakade S M, et al. Multi-view clustering via canonical correlation analysis [A]. Bottou L. ICML 2009: Proceedings, Twenty-Sixth International Conference on Machine Learning [C]. New York: ACM Press, 2009. 129 – 136.
- [12] Olcay Kursun, Ethem Alpaydin, et al. Canonical correlation analysis using within-class coupling [J]. Pattern Recognition Letters, 2011, 32(2): 134 – 144.
- [12] Guha S, Gunopulos D, et al. Correlating synchronous and asynchronous data streams [A]. Lise G. Proceedings of the

- 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03[C]. New York: ACM Press, 2003. 529 – 534.
- [13] 杨雪梅, 董逸生, 等. 高维数据流的在线相关性分析[J]. 计算机研究与发展, 2006, 43(10): 1744 – 1750.  
Yang Xuemei, Dong Yisheng, et al. Online correlation analysis for multiple dimensions data streams[J]. Journal of Computer Research and Development, 2006, 43(10): 1744 – 1750. (in Chinese)
- [14] 王永利, 徐宏炳, 等. 基于低阶近似的多维数据流相关性分析[J]. 电子学报, 2006, 34(2): 293 – 300.  
WANG Yong-li, XU Hong-bing, et al. A correlation analysis algorithm based on low-rank approximation for multiple dimension data streams[J]. Acta Electronica Sinica, 2006, 34(2): 293 – 300. (in Chinese)
- [15] Wang Yongli, Zhang Gongxuan, et al. ApproxCCA: an approximate correlation analysis algorithm for multidimensional data streams[J]. Knowledge-Based Systems, 2011, 24(7): 952 – 962.
- [16] 彭岩, 张道强. 半监督典型相关分析算法[J]. 软件学报, 2008, 19(11): 2822 – 2832.  
Peng Y, Zhang DQ. Semi-supervised canonical correlation analysis algorithm[J]. Journal of Software, 2008, 19(11): 2822 – 2832. (in Chinese)
- [17] 顾晶晶, 陈松灿, 等. 用局部保持典型相关分析定位无线传感器网络节点[J]. 软件学报, 2010, 21(11): 2883 – 2891.  
GU Jing-jing, CHEN Song-can, et al. Localization in wireless sensor network using locality preserving canonical correlation analysis[J]. Journal of Software, 2010, 21(11): 2883 – 2891. (in Chinese)
- [18] DeGroat R D, Roberts R A. SVD update algorithms and spectral estimation application[A]. Donald E K. Nineteenth Asilomar Conference on Circuits, Systems & Computers[C]. Washington: IEEE Computer Society Press, 1985. 601 – 605.
- [19] 张贤达. 矩阵分析与应用[M]. 北京: 清华大学出版社, 2004. 630 – 637.
- [20] Sherman J, Morrison W J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix[J]. Ann Math Statist, 1950, 21(1): 124 – 127.
- [21] Gene H Golub, Charles F Van Loan. Matrix Computations (third edition)[M]. Baltimore and London: The Johns Hopkins University Press, 1996. 461 – 467, 341 – 350.
- [22] 高惠璇. 应用多元统计分析[M]. 北京: 北京大学出版社, 2005. 41 – 44.
- [23] Peters G, Wilkinson J H. Eigenvalues of  $Ax = Bx$  with band symmetric A and B[J]. The Computer Journal, 1969, 12(4): 398 – 404.

## 作者简介



**杨 静** 女, 1962 年 2 月出生于黑龙江省哈尔滨市. 现为哈尔滨工程大学计算机科学与技术学院教授、博士生导师. 主要研究方向为数据库与知识工程、数据挖掘技术、隐私保护、软件理论等.

E-mail: yangjing@hrbeu.edu.cn



**李文平** 男, 1979 年 10 月出生于贵州省大方县. 现为哈尔滨工程大学计算机科学与技术学院博士研究生. 研究方向为数据库与知识工程、软件理论、数据挖掘技术、隐私保护.

E-mail: liwenping@hrbeu.edu.cn



**张健沛** 男, 1956 年 11 月出生于哈尔滨市. 现为哈尔滨工程大学计算机科学与技术学院教授、博士生导师. 主要研究方向为数据库理论与应用、数据挖掘技术、软件理论、社会网络等.

E-mail: zhangjianpei@hrbeu.edu.cn