基于低阶近似的多维数据流相关性分析

王永利[1,2], 徐宏炳[1], 董逸生[1], 钱江波[1], 刘学军[1]
(1.东南大学计算机科学与工程系, 江苏南京 210096; 2.佳木斯大学公共计算机教研部, 黑龙江佳木斯 154007)

摘 要: 目前存在的多数据流相关性分析方法大多只针对于单属性维数据流, 无法体现多变量组成的场与场之间真实的相关性. 为了在资源受限的环境下快速检测多维数据流之间的相关性, 本文提出一种新颖的基于典型相关性分析(CCA)的多维数据流相关性分析算法 StreamCCA, 针对传统的 CCA 计算中的性能瓶颈, 提出为样本方差阵与协差阵组成的乘积阵降维的高效低价近似方法, 在保持分析精度的前提下显著地提高了计算效率. 经理论分析和实验证明, StreamCCA 能够在线精确地识别两条多维数据流的相关关系, 可以作为通用的预报和诊断分析工具广泛应用于数据流挖掘领域.

关键词: 数据流; 典型相关性分析; 低阶近似; 不等概采样; 数据流挖掘
中图分类号: TP311 文献标识码: A 文章编号: 0372-2112 (2006) 02-0293-08

A Correlation Analysis Algorithm Based on Low-Rank Approximation for Multiple Dimension Data Streams

WANG Yong-li[1,2], XU Hong-bing[1], DONG Yi-sheng[1], QIAN Jiang-bo[1], LIU Xue-jun[1]
(1. Department of Computer Science & Engineering, Southeast University, Nanjing, Jiangsu 210096, China;
2. Department of Common Computer Teaching, Jiamusi University, Jiamusi, Heilongjiang 154007, China)

Abstract Presently existing correlation analysis method for multiple data streams were all oriented single dimensions data streams only, which could not identify the real correlation between fields built by multiple variables. To quickly detect correlations between two multiple dimension data streams under constrained resources, a novel correlation analysis algorithm based on canonical correlation analysis (CCA), called StreamCCA, is proposed. Focusing on the computational bottleneck of traditional CCA, StreamCCA introduces a low-rank approximation technique to reduce the dimensionality of product matrix resulted from sample correlation matrix and sample variance matrix, which improves computational performance efficiently on the premise of holding approximate precision. Theoretic analysis and experiments results on synthetic and real data sets indicate that StreamCCA can online detect correlations between multiple dimension data streams accurately. The algorithms proposed herein, are presented as generic forecasting and diagnosis tools, with a multitude of applications on data streams mining problems.

Key words data streams; canonical correlation analysis; low-rank approximation; non-equal probability sampling; data streams mining

1 引言

多维数据流相关性分析在股票趋势预测, 高速网络故障诊断, 天气预报等许多需要在线趋势分析的领域具有广泛的应用. 例如在传感器网络中这相当于分析场与场之间的相关或耦合关系. 如图 1所示, 区域 X 中部署了 p 个传感器, 区域 Y 中部署了 q 个传感器, 分别感知 X 和 Y 两个不同事件源的信息, 假设时间窗口长度为 n, 传感器网络

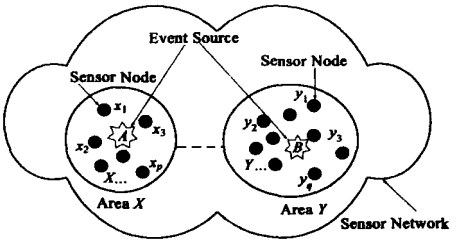

图 1 传感器网络中多维数据流相关性分析

收稿日期: 2005-08-21; 修回日期: 2005-11-27
基金项目: 江苏省 2004年度研究生创新计划项目 (No. xm04-236); 江苏省高技术项目 (No. BG2004034)

具有统一的采样时钟, 在某个采样时刻, 区域 X 的 p 个传感器相当于 p 个维, 类似的区域 Y 采集了 q 维数据, 问题是怎样借助两个传感数据集判断事件源 X 和 Y 是否相关? 如果相关, 又是哪些传感器感知的信息占有主导作用?

多维数据流之间的相关性分析属于统计类的连续查询, 其特点是: 数据流的维数通常很高; 需要频繁计算高维矩阵的乘、转置、求逆等极为耗时的操作; 数据流产生的数据量理论上是无限的, 无法物化所有的流数据; 用户要求以连续的方式在线给出相关性信息等. 这些特点为传统的相关性统计分析理论提出了挑战. 在计算能力受限的条件下, 用户可以接受具有精度保障的近似解答, 因此牺牲部分准确性来换取速度的数据约简技术是解决此类问题的关键.

2 相关工作

多维数据流和多数据流分析已有一些重要的研究结果. Zhu和 Shasha[1] 提出一种基于时间序列数据流模式的 StatStream 模型, 用于发现在任意时刻相关的两条流. 文中采用基窗口的离散 fourier变换来近似计算流的相关性, 计算相关系数时只考虑了数据流单个属性值. Bulut和 Singh[2] 提出一种多分辨索引模式, 监视不同的时间尺度, 有效地解决了可变长度的查询. 其数据流计算模型是 , , x[i], , , 解决的三种查询聚集、模式监控和相关性查询都是针对单维数据流的, 其中计算两个流序列 x 和 y 的相关性被简化为计算它们之间 z2norm 之内的 Euclidean距离, 属于单变量统计.

StreamSVD[3] 是基于低阶近似理论采用奇异值分解 (SVD)方法的一种快速多数据流相关性分析算法, 主要应用于单属性的多数据流之间的相关性分析. 在多元线性相关分析方面, 典型相关分析 (Canonical Correlation Analy2sis, 简称 CCA)和奇异值分解 (SVD)是目前最先进的具有统计意义的方法, 它们都是把变量场 X 和 Y 分解为两两互不相关的独立变量 L 和 M, 通过分别对应的特征向量 A 和 B 的数值分布, 来确定 X 与 Y 的空间相关分布形式. 然而从求极值的函数来看, SVD 方法是在特征向量正交的约束条件下对组合变量间的协方差求最大. 即 SVD 方法旨在使协方差最大, 而不等价于使相关系数最大. 由于度量单位和尺度的差异, 协方差不是相关性的一个好的度量, 于是由此提取的线性组合无法很好地体现变量之间的相关. 与此相反, CCA 方法是在组合变量正交归一化的条件下对组合变量间的相关系数求最大, 这种相关是清楚的和有意义的, CCA 有利于识别多条高维数据流之间耦合的最优时间模态, 适合于作预报, 也可以用于诊断分析. 因此 Strea2mSVD 不适合直接用于多维数据流之间相关性的分析.

综上所述, 目前数据流挖掘领域中有关相关性分析的工作主要针对单属性数据流, 本质上都是属于单变量统计分析的方法, 无法有效地解决前面示例中判断两个场之间是否相关的问题, 即无法检测两条乃至多条多维数据流之间的相关关系. 由于多变量统计分析通过多个侧面探查事物规律, 可以比单变量统计更为精确地分析问题, 故更具有应用和研究价值. 尽我们所知, 目前数据流研究领域还没有基于多变量统计分析理论分析多维数据流之间相关性的文献出现.

矩阵的低阶近似是有效的高维数据约简技术, 其含义是: 给定矩阵 $A_{m@}$ 找到一个秩至多为 k 的矩阵 D^*, 即使 $+A - D^* +_F^2$ 尽可能小[4]. 低阶近似的另一种描述是, 如果将 A 的行视为 R^n 中的点, 问题是发现一个 k维线性子空间, 能够使这些点之间的平方距离和最小. 为了实现高维数据流的在线低价近似, 需要寻找计算复杂度较低的数据约简方法. 本文基于 CCA 统计理论和近似技术提出一种在线分析高维数据流相关性的方法, 其核心思想是使用增量计算模式和具有精度保证的低阶近似技术提高典型性相关分析的效率.

3 相关定义及理论基础

本节首先给出统计意义上两组数据相关性的定义, 然后给出多维数据流模式上相关性的含义.

定义 1 相关性. 设 $f(t_i), g(t_i)$表示两组数据的函数, t_i 表示单维或多维元素, 如果 $f(t_i)S\ f(t_i) - \frac{1}{N}\sum_{j=1}^{N} f(t_j)$, $R_f S \sqrt{\frac{1}{N-1}\sum_{i=1}^{N} f(t_i)}$, 那么借助 $C(S) S \frac{1}{R_f R_g}\left[\frac{1}{N-1}\sum_{i=1}^{N} f(t_i)g(t_i+S)\right]$ 可以获知两组变化数据的相关性.

当 S= 0时属于实时比较, 本文重点研究此种情况. 若 C(S= 0) = 1则两组数据变化相同, 表示强相关记为 $f(t_i)$ Wg(t_i); 若 C(S= 0) = - 1则两组数据变化相反, 表示强反相关, 记为 $f(t_i)$ W- $g(t_i)$; C(S= 0) = 0若则两组数据变化毫不相同. 当 SX0时属于非实时比较, 特别地当 f= g 时, 可以发现数据的自相关性, 能够揭示数据变化的周期性.

311 多维数据流模式

我们在 Turnstile模式[5]的基础上, 定义适合于多维数据流分析的滑动数据流窗口模式.

多维数据流 , a_i, 可定义为多维信号 X 到实数集上的一个映射: X[1, N] y R^p, 每个 a_i 是对 X[j]更新的值. a_i = (j, $\$_i$)其含义为 $X_i[j] = X_{i2l}[j] + \$_i$, $\$_i$ 可能为正也可能为负, 表示在时刻 t的 p 维更新向量, 其中每个分量 $\$_{i,t}$ (i= 1, 2, , p)表示一个属性的更新值. 向量 $\$_i$ 只能读取一次, 按照时间戳 t增加的顺序流入. 定义包含最近 n项元素的序列 $a_{t2n+1}, , , a_t$ 为多维数据流的滑动窗口模式, 这样多维数据流可以视为矩阵. 这里只是概念意义上的矩阵, 并不需要真正物化这个矩阵. 向量 $\$_t$ 相当于关系表中的

元组, 下文中统一采用术语元组描述.

令 $X_{n\times p}$ 表示时间窗为 n 的具有 p 个属性的多维流矩阵, $Y_{n\times q}$ 表示时间窗为 n 的具有 q 个属性的多维流矩阵(不失一般性设 $p<=q$), $X_{i,t}$ 表示 X 中第 t 个时刻的第 i 个属性值; $X_{(i)}$ 表示 X 的第 i 行; $X_{(t)}$ 表示 X 的第 t 列. 如果对元组 $X_{(t)}$ 和 $Y_{(t)}$ 的更新没有限制, 那么两条数据流是异步的, 在连续的元组间没有任何协调. 如果在每一个时刻 t, 元组 $X_{(t)}$ 有 p 个值, 元组 $Y_{(t)}$ 有 q 个值, 每个值对应于一个属性值的流入, 我们称这种情况下的 X, Y 是同步的. 元组按时间 t 有序, 如果任一多维数据流在时刻 t 未接收到元组, 则以全 0 的元组代替, 即允许稀疏数据流矩阵存在.

312 多维数据流 CCA

多维数据流 X 与 Y 之间典型相关性分析的基本思路为: 以最大限度地提取 X 与 Y 之间相关关系的主要特征为准则, 从 X 中提取组合变量 U, 从 Y 中提取组合变量 V, 如下式所示:

$$\begin{cases} U_{p\times 1}=X_{p\times n}A_{n\times 1} \\ V_{q\times 1}=Y_{q\times n}B_{n\times 1} \end{cases} \quad (1)$$

式中, A、B 为线性变换, 又称为空间特征向量. 按式(1)把具有较多个变量的数据流矩阵 X 与 Y 之间的相关化为较少组合变量 U 与 V 间的相关. 我们不加证明的给出总体典型相关的主要结果, 详细证明参见文献[6].

定理 1　设 $X=(X_{(1)},X_{(2)},\cdots,X_{(p)})^T$, $Y=(Y_{(1)},Y_{(2)},\cdots,Y_{(q)})^T$, $Var(X)=\Sigma_{11}$, $Var(Y)=\Sigma_{22}$, $Cov(X,Y)=\Sigma_{12}$, 其中 Σ_{11} 和 Σ_{12} 均为满秩阵且 $p\leq q$. 则 X, Y 的第 k 对典型变量为:

$$U_k=e_k^T\Sigma_{11}^{-1/2}X,\ V_k=f_k^T\Sigma_{22}^{-1/2}Y,\ k=1,2,\cdots,p. \quad (2)$$

其典型相关系数为:

$$\rho_{U_k,V_k}=\rho_k,\ k=1,2,\cdots,p \quad (3)$$

其中 $\rho_1^2\geq\rho_2^2\geq\cdots\geq\rho_p^2$ 为 p 阶矩阵 $M=\Sigma_{11}^{-1/2}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1/2}$ 的特征值, $e_1,e_2,\cdots,e_p$ 为相应的正交单位化特征向量, $f_1,f_2,\cdots,f_q$ 为 q 阶矩阵 $N=\Sigma_{22}^{-1/2}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1/2}$ 的对应于前 p 个最大特征值(按由大到小的次序排列)的正交单位化特征向量.

在实际的数据流应用中, 一般 Σ 是未知的, 需要通过样本来估计. 使用长度为 n 的滑动时间窗口中的样本进行典型相关性分析的过程如下:

设 $\begin{bmatrix}X_{(i)}\\Y_{(i)}\end{bmatrix}$ $(i=1,2,\cdots,n)$ 为来自总体 $\begin{bmatrix}X\\Y\end{bmatrix}$ 的一个样本, 其中 $X_i=(X_{1i},X_{2i},\cdots,X_{pi})^T$, $Y_i=(Y_{1i},Y_{2i},\cdots,Y_{qi})^T$, $(i=1,2,\cdots,n)$, 计算 $X_{(i)}$ 的方差阵 $S_{11}=\frac{1}{n-1}\sum_{i=1}^{n}(X_{(i)}-\bar{X})(X_{(i)}-\bar{X})^T$, $Y_{(i)}$, $Y_{(i)}$ 的方差阵 $S_{22}=\frac{1}{n-1}\sum_{i=1}^{n}(Y_{(i)}-\bar{Y})(Y_{(i)}-\bar{Y})^T$, 协差阵 $S_{21}=\frac{1}{n-1}\sum_{i=1}^{n}(X_{(i)}-\bar{X})(Y_{(i)}-\bar{Y})^T=S_{21}^T$, 其中 $\bar{X}=\frac{1}{n}\sum_{i=1}^{n}X_{(i)}$, $\bar{Y}=\frac{1}{n}\sum_{i=1}^{n}Y_{(i)}$, 分别使用 S_{11}, S_{12}, S_{21}, S_{22} 代替 Σ_{11}, Σ_{12}, Σ_{21}, Σ_{22}, 得到样本典型相关变量, 根据 Bartlett 检验确定保留典型变量的对数, 最后得到典型相关系数.

由定理 1 知 CCA 有两处影响整体运行效率的瓶颈: (1)生成 X 和 Y 的样本协差阵 S_{12}、S_{21} 及各自的方差阵 S_{11}、S_{22}, 其时间代价分别为 $O(p^2n)$ $O(q^2n)$ 和 $O(pqn)$; (2)求解 $C=S_{11}^{-1/2}S_{12}S_{22}^{-1}S_{21}S_{11}^{-1/2}$ 的特征值与特征向量, 其时间代价为 $O(p^2)$. 对于在线相关性分析需求上述时间代价是不能被接受的, 由于存在稀疏矩阵并且在每个流矩阵中只是少数几个属性维起决定作用, 因此我们考虑通过不等概采样的低阶近似技术来提高效率.

4 低阶近似理论及算法

为了改善 CCA 两处计算瓶颈的效率, 我们提出一种基于近似技术的快速多维数据流相关性分析算法 Stream2CCA, 以增量更新的方式维护两个数据流样本矩阵的协差阵 S_{21}, S_{22} 和各自的方差阵 S_{11}, S_{12}; 然后对高维的矩阵进行列向和行向上的采样实现维数约简, 降低生成典型相关系数的代价. 本节主要讨论低阶近似的相关问题, 增量计算方法留待第 5 节中给出.

真实数据流通常具有大量属性维(即 p、q 的值很大), 计算复杂度很高, 为了实现实时的相关性分析, 需要使用少量属性维形成的简单矩阵代替原来的数据流矩阵. 如果可以在保证特征值精度的前提下对 C 降维, 则可以降低计算典型相关系数和特征向量的代价. 这是寻找低阶近似的问题: 为矩阵 $C_{p\times p}$ 找到一个秩至多为 k 的矩阵 W^*, 使 $\|C-W^*\|_F^2\leq[\min_{W,\mathrm{rank}(W)\leq k}\|C-W\|_F^2+\|E\|_F^2]$ 以高概率成立.

矩阵 C 的特征值表示相关性的强度, 如果一个特征值比其他的特征值大许多, 那么其对应的特征向量就代表在前 k 个最大相关向量张成子空间中更强的线性相关. 为了便于讨论典型相关性分析的近似精度, 我们首先量化特征值的强度如下:

定义 2　(E间隔特征值)设矩阵 C 的秩为 r, 特征值为 $\lambda_1,\lambda_2,\cdots,\lambda_r$, 不失一般性, 设 $|\lambda_1|\geq|\lambda_2|\geq\cdots\geq|\lambda_r|$, 特征值集合的 E间隔值是使不等式 $|\lambda_i|\geq(1+E)|\lambda_{i+1}|$ $(1\leq i\leq r)$ 成立的所有 E $(E\geq 0)$ 中最小的, 对于这样的 E 我们说这种特征值是 E间隔的.

注意这样的 E 总是存在的, 其大小代表了特征向量在线性组合中所占的重要程度. 如果 E 很小, 特征值在量上很接近, 那么所有的特征向量都很重要, 如果 E 很大, 则在线性组合中沿着最大特征值的特征向量方向的线性组合最重要. 为了实现对矩阵 C 的低阶近似, 我们引入适合于

数据流矩阵的采样方法:

定义 3 (不等概采样). 设 $Z_1, Z_2, \cdots, Z_n$ 是一组概率 $\sum_{i=1}^{n} Z_i = 1$, 按这组概率对总体中 N 个单元进行放回采样, 每次抽中第 i 个单元的概率为 Z_i, 独立地进行 n 次这样的采样, 称这种采样方法为不等概采样. 如果某个单元的大小或规模的测度为 C_i, 则 Z_i 可以取 $Z_i = C_i / \sum_{i=1}^{n} C_i$.

如果视数据流矩阵中的行 (或列)为采样单元, 则可以实现矩阵不等概行 (或列)采样. 根据矩阵论[7], 本文采用 Frobenius范数和 2-范数作为矩阵和行 (或列)的重要程度的测度, 我们取第 i 行 (或列)的概率为 $Z_i = A\|X_{(i)}\|_2^2 / \|X\|_F^2$ (A 为任意的实数且 $0 < A < 1$), 根据 $\{Z_i\}$ 的缩放拾取列的数目, 概率 $\{Z_i\}$ 保证以更大的可能取到更为重要的行 (或列), 缩放因子 A 可以认为是对过于沉重行 (或列)的修正.

Johnson-Lindenstrauss引理[8]是对高维矩阵进行维数约简的有效技术, 能够在确定的精度内保证约简的质量.

引理 1 (JL 引理)在空间 R^n 中给定包含 n 个向量的集 V, 如果存在矩阵 $S \in R^{s \times n}$, $s = O(1/\varepsilon^2 \log n)$, 其中每个元素 S_{ij} 取自于高斯分布, 并能够适当地缩放, 那么对任何向量 $x \in V$, 不等式 $(1-\varepsilon)\|x\|_2 \le \|Sx\|_2 \le (1+\varepsilon)\|x\|_2$ 可以以高概率 $O(1/n)$ 成立.

证明见文献[8]. JL 引理表明如果利用矩阵对一个 n 维向量进行变换, 根据高斯分布选取矩阵的元素, 则可以以高概率在 s 维的结果空间中保持向量间的相对距离. 我们基于 JL 引理和定义 3中的不等概采样方法在 Euclidean 空间中对矩阵 C 降维: 先对 C 实施行采样, 然后对 C 实施列采样, 根据 JL 引理确定采样行 (或列)的数目, 并结合随机高斯概率和每一行 (或列)的测度选取行 (或列). 采样的概率由下面引理确定:

引理 2 设 P_i 为对矩阵 C 实施不等概行采样形成矩阵 D 时每一行选中的概率, P_{cj} 为继续对 D 进行不等概列采样时的每一列选中的概率, 当 $P_i = A\|C_{(i)}\|_2^2 / \|C\|_F^2$, $i \in \{1, 2, \cdots, p\}$ 时, 如果 P_{cj} 满足不等式 $P_{cj} \ge c/2 + \|D_{(j)}\|_2^2 / \|D\|_F^2$, $j \in \{1, 2, \cdots, p\}$, 则 P_{cj} 与 P_i 符合相同的不等概采样测度.

证明 根据不等概采样定义, 上述行采样兼列采样的过程可等价于先按 P_i 从 C 选出某一行, 然后按条件概率 $Q_{j/i} = \frac{P_{ij}}{P_i}$ (其中 $P_{ij} = C_{(i,j)}^2 / \|C\|_F^2$) 选择 $D^{(i)}$ 中的第 j 项, 于是

$$P_{cj} = \sum_{i \in \{1,\cdots,p\}} \frac{Q_{j/i}}{p} = \sum_{i \in \{1,\cdots,p\}} \frac{AP_{(i,j)}}{pP_i} = \sum_{i \in \{1,\cdots,p\}} \frac{AC_{(i,j)}^2}{pP_i \|C\|_F^2} = \frac{A}{\|C\|_F^2} \sum_{i \in \{1,\cdots,p\}} \frac{C_{(i,j)}^2}{pP_i} = A\|D_{(j)}\|_2^2 / \|D\|_F^2 \le \frac{A}{2} + \|D_{(j)}\|_2^2 / \|D\|_F^2.$$

证毕.

根据引理 1与引理 2给出对 C 进行维数约简的算法:

Procedure Sample_Row_And_Column(C, p, A, E, W)

(1) $s = O(1/\varepsilon^2 \log_2 p)$; k = 0; r = 0; //确定采样的行数与列数, k 与 r 为更新索引

(2) for(i = 0; i < p; i++) begin

(3) Randomly generate a value $G_i \sim N(0, 1)$;

(4) if ($G_i < A + C^{(i)}\|_2^2 / \|C\|_F^2$) then $D^{(k)} = C^{(i)}$; k++; $\|D\|_F^2 = \|D\|_F^2 + C^{(i)}$ //$\|D\|_F^2$ 的初值为 0

(5) if (k > s) then exit for;

(6) end for

(7) for (j = 0; j < p; j++) begin

(8) Randomly generate a value $G_j \sim N(0, 1)$;

(9) if ($G_j < A/2 + D_{(j)}\|_2^2 / \|D\|_F^2$) then $W_{(r)} = D_{(j)}$; r++;

(10) if (r > s) then exit for;

(11) end for

(12) Output W;

采样需要的 $\|C\|_F^2$ 和 $\|C^{(i)}\|_2^2$ 可以在形成 C 的时候一并生成, 第 9步中需要的 $\|D\|_F^2$ 在第 4步已经获得, 只要计算 $\|D_{(j)}\|_2^2$, 其时间代价为 O(s), 因此采样形成 W 的时间复杂度为 O(ps).

下面证明矩阵 C 的列采样兼列采样矩阵 W 与矩阵 C 在 Frobenius范数的意义上是好的近似.

引理 3 对于按照上述步骤构造的矩阵 D 与 W, 不等式 $1/2 + \|C\|_F^2 \le \|D\|_F^2 \le 3/2 + \|C\|_F^2$ 和 $1/2 + \|D\|_F^2 \le \|W\|_F^2 \le 3/2 + \|D\|_F^2$ 至少以概率 $1 - 16/A^2 s$ 成立.

证明 $E(\|D\|_F^2) = E\left(\sum_{r=1}^{s} \|D_{(r)}\|^2\right) = \sum_{r=1}^{s} \sum_{i=1}^{p} P_i \frac{C_{i,r}}{sP_i}$

$= E\left(\sum_{r=1}^{p} \|C_{(r)}\|^2\right) = \|C\|_F^2$, 即 $\|D\|_F^2$ 是 $\|C\|_F^2$ 的无偏估计.

对于 D 的任意一行 $D^{(i)}$, $\|D^{(i)}\|_F^2 \le \|C\|_F^2 / As$, 随机变量 $\|D^{(i)}\|_F^2$ 是 s 个相互独立的随机变量的累加和, 因此,

$$\mathrm{Var}(\|D\|_F^2) = s\mathrm{Var}(\|D^{(i)}\|_F^2) \le sE(\|D^{(i)}\|_F^4) \le \frac{1}{A^2 s} \|C\|_F^2$$

根据 Chebychev不等式可以得到第 1部分的证明, 按类似方法可以得到第 2部分的证明, 证毕.

由引理 3可知, 给定矩阵C, 如果分别对其进行多项行采样兼多项列采样, 得到的矩阵W 是 C 的良好的低价近似, 可以达到较小的信息损失. 在这种采样方法下, 需要进一步估计 C 和近似阵W 之间特征值和特征向量的关系. 有如下引理:

引理 4 设 Q 是 C 的典型特征值, Q_c 是W 的典型特征向量, 其中 s @s 维的W 是 C 的行采样兼列采样阵, 如果W 的形成满足 JL 引理, 那么 $(1-\varepsilon)|Q| \le |Q_c| \le (1+\varepsilon)|Q|$.

证明　设 u_1 是 C 的特征值 Q 对应的典型特征向量, u_1 是 W 的特征值 Qc 对应的典型特征向量, 设 $u_1 = (N_1, N_2, \ldots, N_p)^T$, 根据 JL 引理, $\|u_1\|_2 \le (1+E)\|u_1\|_2$, 由于 W_{ik} 的选取符合 JL 引理, 所以 $\|Wu_1\|_2 = \sqrt{\sum_{i=1}^{s}\left[\sum_{k=1}^{p} W_{ik}N_k\right]^2} \le (1+E)\sqrt{\sum_{k=1}^{p} N_k^2} \le (1+E)\sqrt{\sum_{i=1}^{p}\left[\sum_{k=1}^{p} C_{ik}N_k\right]^2} = (1+E)\|Cu_1\|_2$,

于是 $\|Q_c u_1\|_2 \le (1+E)\|Qu_1\|_2$, 根据 $\|\cdot\|_2$ 范数的齐次性有 $|Q_c|\|u_1\|_2 \le (1+E)|Q|\|u_1\|_2$ 成立, 因此 $|Q_c| \le (1+E)|Q|$ 成立, 同理可证 $(1+E)|Q| \le |Q_c|$ 成立, 证毕.

由引理 4得知, 使用近似矩阵 W 生成的最大特征值与原始矩阵 C 生成的最大特征值满足 $(1 \pm E)$ 的近似精度.

5　StreamCCA 算法的实现

根据前面的分析, 我们提出的 StreamCCA 算法可以分为两个阶段: 第 1阶段, 增量计算方差阵和协差阵; 第 2阶段, 对乘积矩阵 C 进行低阶近似处理, 然后求得典型特征值和典型相关特征向量. 借助参数 A D和 E的变化, 算法的性能可以在精度和计算速度之间折中. 首先给出同步多维数据流的增量维护 S_{11}、S_{12}、S_{21}、S_{22} 算法, 作为下面 StreamCCA 算法的基础.

5.1　增量计算 S_{11}、S_{12}、S_{21}、S_{22}

对于滑动窗口数据流模式, 当同步流的当前输入 $T = (t, \$_x, \$_y)$ 到达时, $X_{(t-n+1)} = X_{(t-n+2)}, \ldots, X_{(t-1)} = X_{(t)}$, $X_{(t)} = X_{(new)}$, 时间窗前滚一个时刻, 接受新的元组 $X_{(new)}$. n步的上述赋值操作归结为 $X = X + \$_x$, 同理有 $Y = Y + \$_y$, 一般情况下可认为 $\$_x = X_{(new)} - X_{(t-n+1)}$, $\$_y = Y_{(new)} - Y_{(t-n+1)}$. 为消除量纲的影响, 同步多维数据流增量归一化算法如下:

Procedure Incremental Centering $(X, Y, \$_x, \$_y, n)$

(1) while a new updating tuple $T = (t, \$_x, \$_y)$ arrived do begin

(2) $X_{(t)} += \frac{1}{n}\$_x$; $Y_{(t)} += \frac{1}{n}\$_y$;

(3) $A_{(new)} = X_{(new)} - X_{(t)}$;

(4) Update head pointer and rear pointer of X, discard the oldest tuple $X_{(t-n+1)}$;

(5) $B_{(new)} = Y_{(new)} - Y_{(t)}$;

(6) Update head pointer and rear pointer of Y, discard the oldest tuple $Y_{(t-n+1)}$;

(7) enddo

(8) Output Centered Matrix A, B and Its Updated Magnitude $\$_a$, $\$_b$.

数据流矩阵以循环队列的方式实现, 更新矩阵时 (即时间窗前滚), 用新元组替换掉最旧的元组, 修改队头和队尾指针, 队头指针指向当前最新元组, 队尾指针总是与队头指针相邻.

下面给出增量生成 X, Y 各自的方差阵及 X 与 Y 协差阵的算法, 为方便描述, 算法中略去了常数项 1/n-1.

Procedure Generating Variance Matrix and Covariance Matrix $(X, Y, \$_x, \$_y, n)$

Input $X \in R^{p \times n}$, $Y \in R^{q \times n}$, X, Y 's updating magnitude $\$_x$, $\$_y$, the length of sliding window n.

(1) Incremental Centering$(X, Y, \$_x, \$_y, n)$;

(2) for all non-zero items in column t of A, B $\{j \mid A_{(j,t)} \neq 0, B_{(j,t)} \neq 0\}$ do begin

(3) if $(j \neq i)$ begin

(4) $S_{11(ij)} += \$_a A_{(t)}$; $S_{22(ij)} += \$_b B_{(t)}$; $S_{12(ij)} += \$_a B_{(t)} + \$_b A_{(t)}$;

(5) end if

(6) if $(j = i)$ begin

(7) $S_{11(ij)} += 2\$_a A_{(t)} + \$_a^2$; $S_{22(i,j)} += 2\$_b B_{(t)} + \$_b^2$; $S_{12(i,j)} += \$_a B_{(t)} + \$_b A_{(t)} + \$_a \$_b$;

(8) end if

(9) $A_{(t)} += \$_a$; $B_{(t)} += \$_b$;

(10) enddo

(11) Output $S_{11} \in R^{p \times p}$, $S_{12} \in R^{p \times q}$, $S_{21} \in R^{q \times p}$, $S_{22} \in R^{q \times q}$

对于矩阵 S_{11}、S_{12}、S_{21}、S_{22} 的更新是以增量方式进行, 生成每个元组的时间代价为 $O(p^2)$, 因此可以以非常有效的方式更新.

5.2　近似 CCA 算法

如果相邻两个元组 (或元组的更新) 到达的时间间隔较长, 可以在每次流更新的时候重新计算 CCA. 然而在更一般的情况下, 数据流从某个特定的计算 CCA 的时刻之后没有明显的变化, 那么就没有必要频繁地重新计算 CCA. 可以证明如果上个时刻的流矩阵与当前时刻的流矩阵几乎处处相同, 那么其特征值和特征向量也处处相同[9]. 所以应该阶段性地计算. 假设最近一次计算 CCA 的时刻为 t_1, 此时同步数据流矩阵为 $Z_1 = \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}$, 观测到元组 $(i, t, \$_{(t)})$ 的当前时刻为 t, 此时同步数据流矩阵为 $Z = \begin{bmatrix} X \\ Y \end{bmatrix}$, 其中 $\$_{(t)}$ 表示 $\begin{bmatrix} \$_{(t)}^x \\ \$_{(t)}^y \end{bmatrix}$, 记从时刻 t_1 到时刻 t 之间 $\$_{(j)}$ 的累加和为 G, 即 $G = \sum_{j=t_1}^{t} |\$_{(j)}|$, G 相当于 $\|Z - Z_1\|_F$[9]. 如果 G 的值没有超过一定阈值, 则 Z_1 对于当前流矩阵 Z 仍然是合理的, 在 t_1 和 t之间不必重新计算 CCA. 设 K_1 是 Z 的最大典型特征值, 由于 $\|Z - Z_1\|_F$ 可以作为 K_1 的估计[9], 因此衡量 G 变化的阈值可以由 K_1 来确定, 为降低频繁计算 CCA 的代价, 可以根据 G 的幅度确定重新计算

CCA 的时机. 如果 G 比 $D\lambda_i$ (D是一个精度系数, 称为跳跃因子)大, 则重新计算, 否则不需重新计算. 如果要求保留多个典型特征值, 则G 应当与 $D(\lambda_1 + \lambda_2 + , + \lambda_k)$相比较. 本文根据文献 [10]提出的一种代价为O (n)的连续计算特征值和特征向量的迭代算法, 通过在能量函数 Rayleigh quotient上的梯度搜索, 计算广义特征值 $\lambda_1, \lambda_2, , , \lambda_k$, 可以避免对大矩阵 Z 进行外积和矩阵转置等代价高昂的运算.

结合前面的理论分析, 下面明确给出同步滑动窗口数据流模式下多维数据流相关性分析算法 StreamCCA, 其工作原理如下所示:

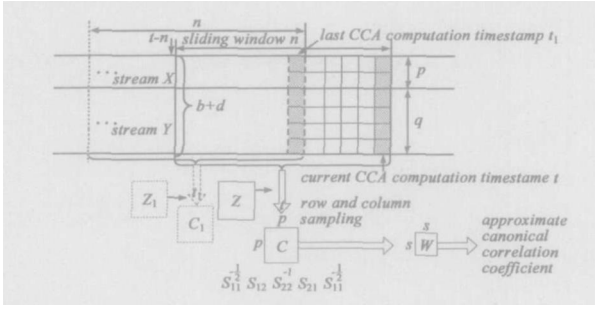

图 2 StreamCCA 算法的工作原理

Procedure StreamCCA (X, Y, ε_x, ε_y, n, A, D, E)

Input 矩阵 $X_{p\times n}$, $Y_{q\times n}$, ε_x, ε_y, 最大特征值误差 E, 当前时间为 S, 设 A= 111

Output 前 k 个最大特征值 (按由大到小的次序排列) 和对应的正交单位化特征向量

(1) while T (i, t, ε_x, ε_y) arrived for (t\ S- n) do begin

(2) Generate Variance Matrix and Covariance Matrix(X, Y, ε_x, ε_y, n)

(3) if (t_1 = t- n) or ($\sum_{j=t_1}^{t} |\varepsilon_{(j)}| |\varepsilon_{(j)}| > D\lambda_1$) begin/* Z 已经过期或 Z_1 矩阵的更新超出阈值* /

(4) $C = S_{11}^{-1/2} S_{12} S_{22}^{-1} S_{21} S_{11}^{-1/2}$;

(5) Sample_Row_And_Column(C, p, A, E, W);

(6)应用W 计算前 k 个最大特征值及对应的正交单位化特征向量;

(7) t_1 = t; /* 设置当前时刻为新的计算 CCA 时刻* /

(8) end if

(9) end do

为实现流式计算, 不必物化整个矩阵 W, 只需保存W 的 hash函数 h[j]集合, 换算式型如 $W_{(j,t)} = h[j](t)$. 根据第 4节中有关近似特征值的推论证明, 算法的第 6步能够保证得到的特征值符合 (1 ? E)近似的要求. 这样矩阵 C 的维数由原来的 p @p 变为 s @s, 因此计算时间复杂度由 $O(p^2)$变为 $O(s^2)$, 由于采样形成W 的额外时间复杂度仅为 O (ps), 所以利用W 计算特征值和特征向量改善了 CCA 计算瓶颈的效率.

6 实验分析

本节描述在合成数据集和真实数据集上对 StreamCCA算法进行的实验评价, 分析不同参数对整体计算精度的影响. 实验计算机的配置是 2 66GHz Pentium /256M / 80G, OS为 Windows 2000, 以 MATLAB 与 Visual C + + 610为工具实现算法. 由于算法应用了基于采样的近似技术, 因此数据集潜在的相关性对于算法的性能有至关重要的影响. 我们选取不同相关度的数据集 X, Y进行测试:

(1)高斯数据集 DS_g: X 中每一维的属性值符合 N_1 (50, 50), Y 中每一维的属性值符合 N_2 (100, 100), 期望其相关系数接近零.

(2)带有噪声的线性数据集 DS_n: 每个属性值取自于线性数据, 然后叠加符合 N (2, 2)的样本, 期望得到较大的相关系数 (> 015).

(3)真实股票数据集 DS_r: 选择标准普尔 500指数 (S&P500)[11]和纳斯达克综合指数 (Nasdaq200)[12]涉及股票的 15年历史数据, 分析两种股票指数的相关性, 用于指导股票的组合投资. 事先无法预计其相关系数的大小, 由实际情况决定.

为度量算法的性能, 定义算法的近似精度 P = 1- $\sqrt{\frac{1}{k}\sum_{i=1}^{k} \varepsilon_Q^2}$ 其中 $\varepsilon_Q = (Q - \hat{Q}) / Q$ 为矩阵 C 的近似特征值的相对误差, $\hat{Q}$ 表示第 i个近似特征值, Q 表示第 i个实际特征值.

611 实验 1. StreamCCA 与 NaiveCCA 运行时间及相关性精度的对比

假设有足够大的缓存空间可以容纳窗口中所有的元组, 称未采用增量与近似技术实现的 CCA算法为 NavieCCA, 图 3中显示了在 (p = 256, q = 512, n = 8192, D= 112, A = 111)条件下 NavieCCA 与 StreamCCA 两种算法在 DS_n 和 DS_r 数据集上每元组平均处理时间的比较, 显而易见, StreamCCA 算法极大地降低了运行时间. 同时我们观察到 StreamCCA 算法的增量计算对于相似性分析的精度并没有显著的影响, 平均精度浮动小于 5%. 分析其原因, StreamCCA 增量计算主要用于构造样本方差阵, 由于样本方差阵形成时与均值和方差有关, 最终用于计算特征值的乘积阵体现的是近一段时间多维数据流的一种总体的统计特征, 因此增量计算对于相似性分析的精度的影响是可以忽略不计的.

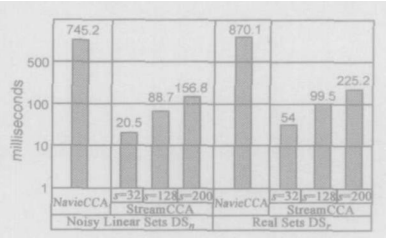

图 3 近似 CCA 与非近似 CCA 运行平均时间对比

图 3中同时给出了选择不同的行 (列) 数 s 时对运行时间的影响, s对应了不同的近似精度参数 E(E=

015, E= 014, E= 012). StreamCCA 在 DS_r 上的运行时间比 DS_n 上的大一些, 是因为有少许时间用于对股票数据按照两种股指的计算方法进行规范化. 在实际应用中, 数据流矩阵的最小维数大于 128 并且参数 E大于 015 时对 C 进行采样才有意义.

612 实验 2 滑动窗口长度 n的变化对近似精度的影响

由于 StreamCCA 涉及到了采样, 在小样本的情况下, 数据集的数据分布情况对近似精度有较大的影响. 首先在 p= 1024, q= 1024, D= 112, E= 0114, n= {2048, 4096, 8192, 16384, 32768}的条件下, 测试 StreamCCA 计算不同数据集的相关系数的近似精度与滑动窗口长度 n 的关系. 测试结果如图 4所示, DS_n 的相关系数近似精度最高; DS_r 呈现出一种强相关性, 相关系数的近似精度次之. 由于高斯分布的样本容量较少, 并不能模拟完全随机的情况, 约减时损失的信息较大, 因此 DS_g 的近似精度最低; 总体上不同分布数据集的平均近似精度保持在 0185左右, 性能稳定, 能够满足大多数统计性应用.

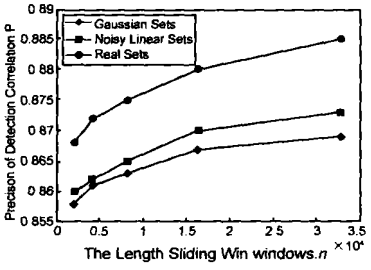

图 4　滑动窗口长度的变化对近似精度的影响

在维数和 E不变的情况下近似阵W 的尺寸保持不变, 随着多维数据流窗口中的数据量 n 的增加, 近似精度呈上升趋势, 这是因为样本数的增加提高了 S_{11}、S_{12}、S_{21}、S_{22}蕴含的信息量, 从而提高了近似精度. 另外在观测中发现, n的变化只影响归一化矩阵的更新代价, 对其他步骤的运行速度影响不大, 每元组平均处理时间的变化并不明显. 这就证明了 StreamCCA 算法的每元组平均处理时间相对恒定, 与窗口大小关系不大, 适合处理无限的流数据.

613 实验 3 维数和参数 D, E的变化对近似精度和运行时间和空间的影响

根据第 4节的分析, 因为求解 C 的特征值与特征向量的时间代价为 $O(p^2)$, 所以维数直接影响 StreamCCA 算法的执行时间. 在 (n= 8192, D= 112, A= 111, E= 0125)的条件下, 使用 DS_n 数据集的 5种不同 (p, q)维数进行实验, 结果证明, StreamCCA 的内存消耗随维数的提高而提高, 主要受维度的影响, 受滑动窗口大小的影响不大. 由于采样的行数 s与列数 s根据可用内存的最大量可以预先设定, 因此主存中常驻的概要 (即约简矩阵)的规模 $O(s^2)$是相对恒定的.

参数的选取对近似算法的性能具有很大的影响, 在 DS_n (p= 1024, q= 1024, n= 8192, A= 111)上测试不同取值的 D与 E的效果. 实验结果证明对于同样的跳跃因子 D,相关系数的近似精度随着 E的增大而呈 Log趋势降低, 这是与低价近似矩阵 W 的尺寸由 E决定, 且 W 的尺寸越小精度越低的推断相吻合的.

7 结束语

经典的统计技术在数据流相关性分析领域中仍然具有无法替代的作用, 本文首次将统计理论中经典的典型相关性分析方法 (CCA)应用于数据流挖掘领域, 从理论上证明了基于不等概采样的低阶近似技术应用于高维数据流复杂计算的可行性, 并且实现了适于数据流处理的多变量相关性分析的增量计算方法. 通过理论分析、实例测试说明, 文中提出的 StreamCCA 技术可以快速精确地分析高维数据流之间相关性, 并能够在近似精度和性能之间折衷. 由于多个高维数据流的相关性可以转变为多个一对多的高维数据流相关性分析, 进而转化为分析两条高维数据流之间的相关性, 因此 StreamCCA 经简单拓展, 可以应用于更为广泛的多个高维数据流之间的相关性分析, 对于医疗、战场传感网络等实时趋势分析应用具有重要意义.

参考文献:

[1] Yunyue Zhu, Dennis Shasha. StatStream: Statistical monitoring of thousands of data streams in real time[A]. Proceedings of 29th International Conference on Very Large Data Bases (VLDB 2002) [C]. Hong Kong, China: Springer-Verlag New York, Inc, 2002. 358- 369.

[2] Ahmet Bulut, Ambuj K. Singh. A unified framework for monitoring data streams in real time[A]. Proceedings of the 21st International Conference on Data Engineering (ICDE 2005) [C]. Tokyo, Japan: IEEE Computer Society, 2005. 44- 55.

[3] Sudipto G, Dimitrios G, Nick K. Correlating synchronous and asynchronous data streams[A]. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD2003) [C]. Washington, USA: ACM Press, 2003. 529- 534.

[4] Achlioptas D, McSherry F. Fast computation of low rank approximations[A]. Proceedings of the 33rd Annual Symposium on Theory of Computing (STOC2001) [C]. Crete, Greece: ACM Press, 2001. 611- 618.

[5] Muthukrishnan S. Data streams: algorithms and applications[A]. Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms [C]. Baltimore, Maryland, USA: Society for Industrial and Applied Mathematics, 2003. 413- 413.

[6] 徐仲, 张凯院, 陆全, 冷国伟. 矩阵论简明教程 [M]. 北京: 科学出版社, 2002. 24- 27.

[7] Johnson R A, Wichern D W. Applied Multivariate Statistical Analysis (3rd Edition) [M]. Prentice Hall Inc,

1992. Chapter 4.
[8] Johnson W. B., Lindenstrauss J. Extension of lipshitz mapping into hilbert space[J]. Contemporary Mathematics, 1984, 26(5): 189-206.
[9] Alan F., Ravi K., Santosh V. Fast monte-carlo algorithms for finding low-rank approximations[A]. Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS98)[C]. Palo Alto, California, USA: IEEE Computer Society, 1998. 370-378.
[10] Borga M. Learning Multidimensional Signal Processing[D]. Sweden: Linkêping University, Sweden, 1998.
[11] S&P500 historical stock exchange data[DB/OL]. http://kumo.swcp.com/stocks, 20052072162005272227.
[12] Nasdaq historical stock exchange data[DB/OL]. http://www.eoddata.com/download.asp, 20052072162005272227.

作者简介:

王永利 男, 1974年出生, 讲师, 博士研究生, 主要研究方向为现代数据管理技术, 数据流处理, 数据挖掘. E-mail: wyl_seu@126.com.

徐宏炳 男, 1947年出生, 教授, 主要研究方向为信息系统的体系结构, ASIC的开发与应用, 实时数据采集与信息处理, 数据流管理系统等. E-mail: hbxu@seu.edu.cn

董逸生 男, 1940年出生, 教授, 博士生导师, 主要研究方向为现代数据管理技术, 包括 XML数据管理技术、语义 web、数据网格、数据仓库、现代信息检索和移动数据库等; 信息系统的建模、体系结构和开发方法等. E-mail: ysdong@seu.edu.cn.