文章编号:1001 - 2486(2004)02 - 0062 - 04

连续求解样本协方差矩阵迹的快速递推算法*

黄晓斌,万建伟,王 展,欧建平
(国防科技大学电子科学与工程学院,湖南 长沙 410073)

摘 要:为解决当样本数据不断增加时,利用传统方法反复计算样本协方差迹耗时多的缺点,提出了一种快速递推算法。理论分析和仿真试验都表明,算法的时间复杂度比传统的方法降低了一个数量级,从而大大减少了计算时间。

关键词:协方差矩阵;时间复杂度;算法
中图分类号:TP391.41 文献标识码:A

A Fast Recursive Algorithm for Continuously Computing the Trace of the Sample Covariance Matrix

HUANG Xiao-bin, WAN Jian-wei, WANG Zhan, OU Jian-ping
(College of Elecnonic Science and Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: In order to resolve the disadvantage that much time is consumed for computing the trace of the sample covariance matrix with the traditional method when the samples continuously increase, a fast recursive algorithm is presented. The theoretical analysis and the simulation results all show that compared with the traditional method, the consuming time with our method is decreased by one order of magnitude, so it can reduce a great deal of computing time.

Key words: covariance matrix; time complexity; algorithm

由于样本协方差矩阵的迹能够描述模式类别的内聚性,因而被知识工程领域中的许多研究分支所采用,如数据挖掘、模式识别[1,2]等。通常情况下,样本数据是不断增加的,如果使用传统的方法来反复计算样本协方差矩阵的迹将非常耗时,因此,有必要寻找一种计算它的快速算法。

1 计算协方差矩阵迹的快速递推算法

设有如下形式的数据矩阵:

$$\begin{cases} \boldsymbol{X}^{(1,N)} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_l, \cdots, \boldsymbol{x}_N] \\ \boldsymbol{X}^{(N+1,N+P)} = [\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+2}, \cdots, \boldsymbol{x}_{N+P}] \\ \boldsymbol{X}^{(1,N+P)} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N, \boldsymbol{x}_{N+1}, \cdots, \boldsymbol{x}_{N+P}] \end{cases}$$

其中,$\boldsymbol{X}^{(1,N)}$表示原始数据矩阵,$\boldsymbol{X}^{(N+1,N+P)}$表示增量数据矩阵,$\boldsymbol{X}^{1,(N+P)}$表示增加样本后的数据矩阵,$\boldsymbol{x}_l = (x_{l,1}, x_{l,2}, \cdots, x_{l,k}, \cdots, x_{l,M})^{\mathrm{T}}$。令 $\boldsymbol{\mu}^{(1,N)} = (\mu_1^{(1,N)}, \mu_2^{(1,N)}, \cdots, \mu_k^{(1,N)}, \cdots, \mu_M^{(1,N)})^{\mathrm{T}}$ 为 $\boldsymbol{X}^{(1,N)}$的均值向量,$T^{(1,N)}$为 $\boldsymbol{X}^{(1,N)}$协方差矩阵的迹;$\boldsymbol{\mu}^{(1,N+P)} = (\mu_1^{(1,N+P)}, \mu_2^{(1,N+P)}, \cdots, \mu_k^{(1,N+P)}, \cdots, \mu_M^{(1,N+P)})^{\mathrm{T}}$ 为 $\boldsymbol{X}^{(1,N+P)}$的均值向量,$T^{(1,N+P)}$为 $\boldsymbol{X}^{(1,N+P)}$协方差矩阵的迹;$\boldsymbol{\mu}^{(N+1,N+P)} = (\mu_1^{(N+1,N+P)}, \mu_2^{(N+1,N+P)}, \cdots, \mu^{(N+1,N+P)}, \cdots, \mu_M^{(N+1,N+P)})^{\mathrm{T}}$ 为 $\boldsymbol{X}^{(N+1,N+P)}$的均值向量。则:

$$\mu_k^{(1,N+P)} = \frac{1}{N+P}\sum_{l=1}^{N+P} x_{l,k} = \frac{1}{N+P}\sum_{l=1}^{N} x_{l,k} + \frac{1}{N+P}\sum_{l=N+1}^{N+P} x_{l,k}$$

$$= \frac{N}{N+P}\left(\frac{1}{N}\sum_{l=1}^{N} x_{l,k}\right) + \frac{P}{N+P}\left(\frac{1}{P}\sum_{l=N+1}^{N+P} x_{l,k}\right)$$

* 收稿日期:2003 - 09 - 30
作者简介:黄晓斌(1978—),男,博士生。

$$= \frac{N}{N+P}\mu_k^{(1,N)} + \frac{P}{N+P}\mu_k^{(N+1,N+P)} \tag{1}$$

由此可得：

$$\boldsymbol{\mu}^{(1,N+P)} = \frac{N}{N+P}\boldsymbol{\mu}^{(1,N)} + \frac{P}{N+P}\boldsymbol{\mu}^{(N+1,N+P)} \tag{2}$$

另有：

$$\begin{aligned} T^{(1,N+P)} &= \frac{1}{N+P-1}\sum_{l=1}^{N+P}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N+P)})^{\mathrm{T}}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N+P)}) \\ &= \frac{1}{N+P-1}\sum_{l=1}^{N}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N+P)})^{\mathrm{T}}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N+P)}) + \\ &\quad \frac{1}{N+P-1}\sum_{l=N+1}^{N+P}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N+P)})^{\mathrm{T}}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N+P)}) \end{aligned} \tag{3}$$

其中：

$$\begin{aligned} &\sum_{l=1}^{N}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N+P)})^{\mathrm{T}}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N+P)}) \\ &= \sum_{l=1}^{N}\left(\boldsymbol{x}_l - \frac{1}{N+P}\sum_{j=1}^{N+P}\boldsymbol{x}_j\right)^{\mathrm{T}}\left(\boldsymbol{x}_l - \frac{1}{N+P}\sum_{j=1}^{N+P}\boldsymbol{x}_j\right) \\ &= \sum_{l=1}^{N}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N)})^{\mathrm{T}}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N)}) + \frac{NP^2}{(N+P)^2}(\boldsymbol{\mu}^{(1,N)} - \boldsymbol{\mu}^{(N+1,N+P)})^{\mathrm{T}}(\boldsymbol{\mu}^{(1,N)} - \boldsymbol{\mu}^{(N+1,N+P)}) \end{aligned} \tag{4}$$

$$\begin{aligned} &\sum_{l=N+1}^{N+P}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N+P)})^{\mathrm{T}}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N+P)}) \\ &= \sum_{l=N+1}^{N+P}\left(\boldsymbol{x}_l - \frac{1}{N+P}\sum_{j=1}^{N+P}\boldsymbol{x}_j\right)^{\mathrm{T}}\left(\boldsymbol{x}_l - \frac{1}{N+P}\sum_{j=1}^{N+P}\boldsymbol{x}_j\right) \\ &= \sum_{l=N+1}^{N+P}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N)})^{\mathrm{T}}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N)}) - \frac{2P^2}{N+P}(\boldsymbol{\mu}^{(1,N)} - \boldsymbol{\mu}^{(N+1,N+P)})^{\mathrm{T}}(\boldsymbol{\mu}^{(1,N)} - \boldsymbol{\mu}^{(N+1,N+P)}) \\ &\quad + \frac{P^3}{(N+P)^2}(\boldsymbol{\mu}^{(1,N)} - \boldsymbol{\mu}^{(N+1,N+P)})^{\mathrm{T}}(\boldsymbol{\mu}^{(1,N)} - \boldsymbol{\mu}^{(N+1,N+P)}) \end{aligned} \tag{5}$$

将(4)、(5) 式代入(3) 式得：

$$\begin{aligned} T^{(1,N+P)} &= \frac{1}{N+P-1}\sum_{l=1}^{N}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N)})^{\mathrm{T}}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N)}) + \frac{1}{N+P-1}\sum_{l=N+1}^{N+P}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N)})^{\mathrm{T}}(\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N)}) \\ &\quad + \frac{1}{N+P-1}\left[\frac{NP^2}{(N+P)^2} - \frac{2P^2}{N+P} + \frac{P^3}{(N+P)^2}\right](\boldsymbol{\mu}^{(1,N)} - \boldsymbol{\mu}^{(N+1,N+P)})^{\mathrm{T}}(\boldsymbol{\mu}^{(1,N)} - \boldsymbol{\mu}^{(N+1,N+P)}) \\ &= \frac{N-1}{N+P-1}T^{(1,N)} + \frac{1}{N+P-1}\left(\sum_{l=N+1}^{N+P}\|\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N)}\|^2 - \frac{P^2}{N+P}\|\boldsymbol{\mu}^{(1,N)} - \boldsymbol{\mu}^{(N+1,N+P)}\|^2\right) \end{aligned} \tag{6}$$

令 $\Omega = \sum_{l=N+1}^{N+P}\|\boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N)}\|^2 - \frac{P^2}{N+P}\|\boldsymbol{\mu}^{(1,N)} - \boldsymbol{\mu}^{(N+1,N+P)}\|^2$**,则计算协方差矩阵迹的递推公式为：**

$$\begin{cases} \mu_k^{(N+1,N+P)} = \dfrac{1}{P}\displaystyle\sum_{l=N+1}^{N+P}x_{l,k} \\ \boldsymbol{\mu}^{(1,N+P)} = \dfrac{N}{N+P}\boldsymbol{\mu}^{(1,N)} + \dfrac{P}{N+P}\boldsymbol{\mu}^{(N+1,N+P)} \\ T^{(1,N+P)} = \dfrac{N-1}{N+P-1}T^{(1,N)} + \dfrac{1}{N+P-1}\Omega \end{cases} \tag{7}$$

初始条件为：

$$\begin{cases} \boldsymbol{\mu}^{(1)} = \boldsymbol{x}_1 \\ T^{(1)} = 0 \end{cases}$$

2 快速递推算法的时间复杂度分析及其与传统计算方法的比较

为使分析问题方便,设定如下的数学模型,并基于此模型来进行时间复杂度分析和仿真试验。

数学模型：记原始数据矩阵为 $X^{(1,N)}$，在此基础上每次增加 P 个样本数据，则第 i 次的增量数据矩阵记为 $X^{(N+(i-1)P+1,N+iP)}$（$i \geqslant 1$），增加后的总数据矩阵记为 $X^{(1,N+iP)}$，共递增 I 次数据，要求依次计算 $T^{(1,N+P)}, T^{(1,N+2P)}, \cdots, T^{(1,N+iP)}, \cdots, T^{(1,N+IP)}$。

采用式(7)来计算 $T^{(1,N+iP)}$，而传统方法由下列三个计算式组成：

$$\begin{cases} \mu_k^{(1,N+iP)} = \dfrac{1}{N+iP}\displaystyle\sum_{l=1}^{N+iP} x_{l,k} \\ \hat{\boldsymbol{x}}_l = \boldsymbol{x}_l - \boldsymbol{\mu}^{(1,N+iP)} \\ T^{(1,N+iP)} = \dfrac{1}{N+iP-1}\displaystyle\sum_{l=1}^{N+iP} \hat{\boldsymbol{x}}_l^{\mathrm{T}}\hat{\boldsymbol{x}}_l \end{cases} \tag{8}$$

表1给出了两种方法 $T^{(1,N+iP)}$ 的计算量。

如果使用汇编语言在 ADSP2191 上实现，则加、减、乘法指令都是一个时钟周期 τ(6.5ns)[3]，除法指令视不同情况而不同，不失一般性，这里用 $\theta\tau(\theta \geqslant 2)$ 表示，其中 θ 为除法指令在各种情况下所用最大时钟周期数目，显然，在某个特定的平台上，它是一个常数。由此可得传统算法计算 $T^{(1,N+P)}, T^{(1,N+2P)}, \cdots, T^{(1,N+iP)}, \cdots, T^{(1,N+IP)}$ 所需时间为：

表1 传统算法与本文算法计算的计算量比较

Tab.1 Computation compared between traditional algorithm and our algorithm

操作	计算量	
	传统算法	本文算法
加法次数 n_a	$2M(N+iP)-M-1$	$2PM+M$
减法次数 n_b	$M(N+iP)$	$PM+M+1$
乘法次数 n_c	$M(N+iP)$	$PM+2M+1$
除法次数 n_d	$M+1$	$2M+2$

$$\begin{aligned} \lambda_1 &= \sum_{i=1}^{I}\left[(n_a+n_b+n_c)\tau + n_d\theta_\tau\right] = \sum_{i=1}^{I}\left[4M(N+iP)+(\theta-1)M+\theta-1\right]\tau \\ &= 2PMI^2\tau + \left[4MN+2PM+(\theta-1)M+\theta-1\right]I\tau \end{aligned} \tag{9}$$

本文算法所需时间为：

$$\lambda_2 = \sum_{i=1}^{I}\left[(n_a+n_b+n_c)\tau + n_d\theta\tau\right] = \left[4PM+4M+2+(2M+2)\theta\right]I\tau \tag{10}$$

显然，当 $I \gg 1, P \gg 1$ 且 $PI^2 \gg N$ 时，传统算法的时间复杂度为 $O(PMI^2)$，本文算法的时间复杂度为 $O(PMI)$，$\lambda_1 \approx 2PMI^2\tau$，$\lambda_2 \approx 4PMI\tau$。对于递增次数 I 来说，本文算法比传统算法提高了一个数量级。

3 仿真实验及分析

固定递增次数 $I=200$，原始数据样本个数 $N=1000$，分别使用 $M=2,5,10,20$；$P=100,500,1000,2000,5000$ 来随机产生递增数据矩阵，并且在每个参数配对(M,P)情况下都独立进行10次试验，对所取得的运算时间求平均值。使用本文算法和传统算法的运算时间对比如表2。

表2 各种增量数据集情况下 λ_1 与 λ_2 对比表(单位：s)

Tab.2 Comparison between λ_1 and λ_2 in each increment database(unit：s)

	传统算法				本文算法			
单次递增	维数				维数			
样本数	$M=2$	$M=5$	$M=10$	$M=20$	$M=2$	$M=5$	$M=10$	$M=20$
$P=100$	1.0203	2.1564	3.8938	7.2483	0.3188	0.3266	0.3266	0.3437
$P=500$	6.2454	12.3330	22.2515	42.1562	1.5360	1.5767	1.6031	1.7094
$P=1000$	13.2125	25.6750	46.3970	89.2546	3.0500	3.1295	3.1999	3.4625
$P=2000$	26.3359	50.8187	91.6735	438.4562*	6.0717	6.2656	6.5173	6.9719
$P=5000$	67.9295	142.7420	–	–	15.5108	15.8687	16.7126	33.6750*

注：试验使用 Matlab 语言，在 Windows 2000 操作系统、Pentium4 处理芯片上实现，所得结果与理论分析可能有出入，仅供参考；“–”表示由于运行时间过长，未得试验结果

从表 2 可以看出，当 M 固定、P 变化时，传统算法和本文算法的运算时间基本上都随 P 线性增长（其中带 * 号的数据不满足此规律，原因在于随着 M 和 P 的增大，参与计算的数据量增大，存储数据将花费额外的时间）。当 P 固定、M 变化时，传统算法的运算时间随 M 的增大，基本上服从线性关系，而本文算法基本上不随 M 变化（理论上应该与 M 呈线性关系），其原因在于 Matlab 语言对向量和矩阵的运算进行了优化，在某一维数范围内，运算时间基本上不随维数的增加而增加，但当参与运算的向量或矩阵的维数超过某一维数时，其运算时间将会增加，但与维数也不呈线性关系。另外，从表中还可明显看出，对于同一个（M，P），传统算法的运算时间明显多于本文算法。

固定维数 $M=2$，同样设定原始样本个数 $N=1000$，分别使用 $I=100,120,140,160,180,200,220,240,260,280,300$；$P=100,200,300$ 来分析两种方法的运算时间。图 1 和图 2 分别给出了两种方法的运算时间与递增次数 I 的关系。

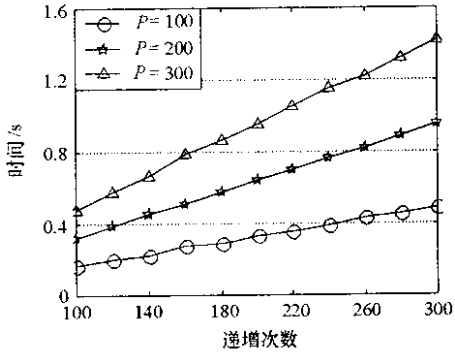

图 1 本文算法随递增次数的性能图（$M=2$）

Fig. 1 Our algorithm performance along with increment number（$M=2$）

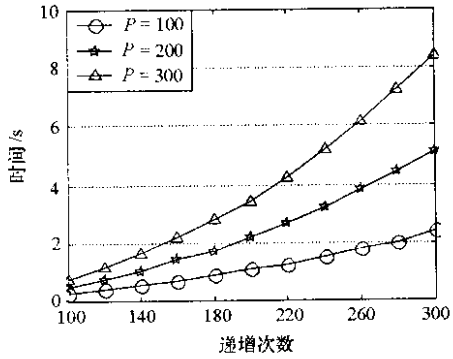

图 2 传统算法随递增次数的性能图（$M=2$）

Fig. 2 Traditional algorithm performance along with increment number（$M=2$）

从图中可以明显看出，传统算法的运算时间与递增次数 I 呈平方关系，而本文算法与递增次数呈线性关系，而且本文算法的运算时间比传统方法明显减少。

4 结束语

本文算法的运算时间在递增次数上比传统算法提高了一个数量级，特别适用于高维、大数据量样本连续递增情况下求取协方差矩阵的迹。提出的快速递推算法有效地减少了反复计算协方差矩阵迹所耗的时间，同时该算法还具有概念清晰、计算简单、易于在 DSP 上实现等一系列优点。

本文的理论分析是基于汇编语言和 DSP 平台的，由于条件所限，仿真试验是在 PC 上使用 Matlab 语言进行的，因此有不一致的情况。使用汇编语言在 DSP 上实现该算法是下一步的工作，同时将应用该算法于模式识别中的某些具体问题来解决时间复杂度。

参 考 文 献：

[1] 边肇祺，张学工，等. 模式识别[M]. 北京：清华大学出版社，1999.

[2] 冯宗哲，程相君. 模式识别原理[M]. 西安：西安电子科技大学出版社，1993.

[3] Analog Devices Inc. ADSP－219x DSP Instruction Set Reference[R]. 1999.