第16卷 第12期
2006年12月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol. 16 No. 12
Dec. 2006

软件度量中主成分分析方法的研究

张 靖,葛 玮,郝克刚
(西北大学 计算机科学系,陕西 西安 710069)

摘 要:文中针对软件度量的数据分析阶段存在大量相关性数据的情况,提出将主成分分析方法引入数据的分析过程。在尽可能多地保留原始数据信息的基础上,使软件度量所涉及的大量相关数据得到降维,筛选出影响被度量对象的主要因素,在降维原始数据的同时使度量的结果更加准确,度量数据的含义更加清晰明确,以达到使整个度量过程更加准确、高效的目的。

关键词:软件度量;主成分分析;特征向量
中图分类号:TP311.5 **文献标识码**:A **文章编号**:1673-629X(2006)12-0144-04

Research of Principal Component Analysis in Software Metrics

ZHANG Jing, GE Wei, HAO Ke-gang
(Department of Computer Science, Northwest University, Xi'an 710069, China)

Abstract: In order to decrease the dimension of mass correlated dataset during data analysis process of software metrics, in the paper, introduce the principal component analysis(PCA) to this period. On the base of keeping as more as possible of the original features of the dataset, successfully lower the dimension of the raw data, and get the main factor which affect the object that we are measured. In this way, make the meaning of measure data clearer and more accurate, and make the whole metrics process more effective.

Key words: software metrics; principal component analysis; eigenvector

0 引 言

软件度量是对软件开发项目、过程及其产品进行数据定义、收集以及分析的持续性定量化过程,目的在于对此加以理解、预测、评估、控制和改善,从而保证软件开发中的高效率、低成本、高质量[1]。度量取向要依靠事实、数据、原理、法则;方法是测试、审核、调查;工具是统计、图表、数字、模型;标准是量化的指标。在软件度量中不可避免地要对大量数据进行合理的处理分析。

软件度量的实质是根据一定规则,对实体属性进行量化表示,从而能够清楚地理解该实体。由于其涉及到软件及其开发过程中方方面面的属性和指标,因此度量时所收集的数据是大量并且复杂的,通常收集到的原始数据表面上杂乱无序,但经过分析变换就能体现出重要规律,因此采用适当的数据分析技术就显得至关重要[2]。数据分析就是在大量实验数据的基础上,也可在正交实验设计的基础上,通过数学处理和计算,揭示软件产品质量和性能指标与众多影响因素之间的内在关系[3]。

收稿日期:2006-03-16
基金项目:国家"八六三"项目(2004AA115090)
作者简介:张 靖(1979-),女,陕西人,硕士研究生,助教,研究方向为软件工程、分布式并行计算;葛 玮,副教授,研究方向为软件工程、分布式计算、工作流技术;郝克刚,教授,博士生导师,研究方向为软件工程、分布式计算、工作流技术。

文中要介绍的主成分分析方法是多元数据分析方法之一,它利用降维的思想,把多变量转化为少数几个综合变量,通过除去变量之间的某些相关性而对数据集进行简化。这种方法主要适用于所考察的大量数据中存在较复杂或较大的相关性的情况。

1 主成分分析方法

1.1 基本思想

在软件度量实施过程中,要对大量的软件属性进行测量,在这些属性中有一些是由相互关联的数据组成的。例如,项目的规模会影响到完成该项目的工作量,因此规模和工作量是相关的。在对项目的持续时间进行预测时,如果在预测持续时间的等式中同时用到了规模和工作量,那么预测出来的时间要比实际需要的时间长,因为在某种意义上进行了重复计算,由规模度量计算出的持续时间可能也由于工作量的值体现了出来[1]。因此一定要确保等式中的变量相互之间尽可能保持独立,主成分分析通过对一组影响某一问题的相关变量进行线性变换,使变换后得到的变量独立不相关,称为综合变量或主成分。这样,主成分不仅保留了原来相关变量中的主要信息,彼此之间又不相关。

设有 n 个被度量的软件模块样本,每个样本有两个观测属性 x_1 和 x_2,在由变量 x_1 和 x_2 所确定的二维平面中,

n 个样本点沿 x_1 轴方向和 x_2 轴方向都具有较大的离散性,其离散的程度可以分别用观测变量 x_1 的方差和 x_2 的方差定量地表示。这种情况下如果只考虑 x_1 和 x_2 中的任何一个,那么包含在原始数据中的度量信息将会有较大的损失;而如果两个都考虑,则在很大程度上进行了度量信息的重复叠加。如果将 x_1 轴和 x_2 轴同时按逆时针方向旋转 θ 角度,得到新坐标轴 y_1 和 y_2,y_1 和 y_2 是两个新变量。根据旋转变换公式 $\begin{cases} y_1 = x_1\cos\theta + x_2\sin\theta \\ y_2 = -x_1\sin\theta + x_2\cos\theta \end{cases}$,这里新变量 y_1 和 y_2 是原变量 x_1 和 x_2 的线性组合,它的矩阵表示形式为:$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \boldsymbol{U}'x$,其中,$\boldsymbol{U}'$ 为旋转变换矩阵,它是正交矩阵,即有 $\boldsymbol{U}' = \boldsymbol{U} - 1, \boldsymbol{U}'\boldsymbol{U} = \boldsymbol{I}$。旋转变换的目的是为了使得 n 个样品点在 y_1 轴方向上的离散程度最大,即 y_1 的方差最大。这样,经过上述旋转变换就可以把原始数据的信息集中到 y_1 轴上,变量 y_1 代表了原始数据的绝大部分信息,对数据中包含的信息起到了浓缩作用。y_1, y_2 除了可以对包含在 x_1, x_2 中的信息起着浓缩作用之外,还具有不相关的性质,这就使得在研究复杂的问题时避免了信息重叠所带来的虚假性。二维平面上的 n 个点的方差大部分都归结在 y_1 轴上,而 y_2 轴上的方差很小。y_1 和 y_2 称为原始变量 x_1 和 x_2 的综合变量。由于 n 个点在 y_1 轴上的方差最大,由此称 y_1 为第一主成分,y_2 为第二主成分[4]。在软件度量研究中,可以只考虑 y_1 方向上的度量信息,忽略 y_2 方向上的度量信息,这样损失的信息不多且无重复计算。

1.2　主成分的数学模型

一般在软件度量中要考虑在多种软件系统中或者多个模块中的一些属性的问题。N 个软件系统或模块就是 n 个样品,而所考察的 p 个属性对则构成了 p 个变量 $x_1, x_2, \cdots, x_p (n > p)$。这样原始统计资料整理的原始数据矩阵为:

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix} \tag{1}$$

通常,一些软件度量中采集到的数据具有不同量纲,有的数据在数量级上也有很大差异,在应用主成分分析研究软件度量问题时,不同的量纲和数量级会引出新的问题。为了消除由于量纲的不同可能带来的一些不合理的影响,在进行主成分分析之前先对数据进行标准化处理,使得每一个变量的平均值为 0,方差为 1。变量标准化的公式为:$x_{ij}^* = \dfrac{x_{ij} - \bar{x}_j}{\sqrt{\mathrm{var}(x_j)}}, (= 1,2,\cdots,n; j = 1,2,\cdots,p)$,其中,$\bar{x}_j$ 和 $\sqrt{\mathrm{var}(x_j)}$ 分别是第 j 个变量的平均值和标准差。为方便,将数据标准化后的矩阵仍用(1)式的 x 记,那么 $x = (x_1, x_2, \cdots, x_p)'$ 的 p 个变量综合成 p 个新变量,新的综合变量可以由原来的变量 $x_1, x_2, \cdots, x_p$ 线性表示,即

$$\begin{cases} y_1 = u_{11}x_1 + u_{12}x_2 + \cdots + u_{1p}x_p \\ y_2 = u_{21}x_1 + u_{22}x_2 + \cdots + u_{2p}x_p \\ \cdots\ \cdots\ \cdots\ \cdots\ \cdots\ \cdots \\ y_p = u_{p1}x_1 + u_{p2}x_2 + \cdots + u_{pp}x_p \end{cases}$$

并且满足 $u_{k1}^2 + u_{k2}^2 + \cdots + u_{kp}^2 = 1\ (k = 1,2,\cdots,p)$,其中,系数 u_{ij} 由下列原则来确定:

(1)y_i 与 $y_j (i \neq j; i,j = 1,2,\cdots,p)$ 相互无关。

(2)y_1 是 $x_1, x_2, \cdots, x_p$ 的一切线性组合中方差最大者;y_2 是与 y_1 不相关的 $x_1, x_2, \cdots, x_p$ 的所有线性组合中方差最大者;以此类推。

如此决定的综合变量 $y_1, y_2, \cdots, y_p$ 分别称为原变量的第一,第二,…,第 p 个主成分。其中 y_1 在总方差中占的比重最大,第一个主成分 y_1 的方差贡献率最大,依次递减,这里第 k 个主成分 y_k 的方差贡献率以 λ_k 表示[4]。

在软件度量中利用主成分分析的目的是为了减少变量的个数,简化系统结构,因此只挑选前几个方差贡献率最大的主成分,而不取所有的 p 个主成分,通常选取 $m < p$ 个主成分。m 的选取一般应使得前 m 个主成分的累积贡献率达到 85% 以上为宜,即 $\sum_{i=1}^{m}\lambda_i(\sum_{i=1}^{p}\lambda_i)^{-1} \geqslant 85\%$,由此确定了要选取的主成分之后,问题便得以简化,在此基础之上,可以进行进一步的分析研究。

2　主成分分析方法的应用研究

在软件可维护性度量中,欧洲软件认证计划的第一阶段从 5 个工业软件项目的模块中收集了 39 个与可维护性相关的软件度量。通过主成分分析将度量减少到 6 个,它们对变差的贡献不到 90%,其中仅规模一项就差不多占了变差的 57%,也就是说,可维护性度量的变差中一半以上都可以由规模度量的改变来说明[1]。

表 1　原始数据集

原始数据集				
样品编号	规模	扇出	扇入	控制流路径
1	29	4	1	4
2	29	4	1	4
3	32	2	2	2
4	33	3	27	4
5	37	7	18	16
6	41	7	1	14
7	55	1	1	12
8	64	6	1	14
9	69	3	1	8
10	101	4	4	12
11	120	3	10	22
12	164	14	10	221
13	205	5	1	59
14	232	4	17	46
15	236	9	1	38
16	270	9	1	80
17	549	11	2	124

文中使用 SPSS 软件辅助分析，现给定一个数据集见表 1，该数据集反映了特定产品子系统内部规程的属性，是一些组件级的数据，包括模块代码行(X_1)、扇出(X_2)、扇入(X_3)和控制路径(X_4)。在研究一些软件度量问题例如分析复杂度或预测工作量与开发成本时，显然这些属性具有一定的相关性。因此，该数据集符合作主成分分析的基本数据特征。

由表 2 知该数据集的中位数比均值小很多。因此，该数据集中的数据不是正态分布的，为了避免量纲的影响，需要将数据集中的数据进行标准化处理，使得每一个变量的平均值为零，方差为 1。标准化后的数据见表 3。

表 2 Descriptive Statistics

	N	Minimum	Maximum	Mean	Median	Std. Deviation
X_1	17	29.0	549.0	133.294	69	135.560
X_2	17	1.0	14.0	5.647	4	3.463
X_3	17	1.0	27.0	5.824	1	7.900
X_4	17	2.0	221.0	40.000	14	57.005
Valid N (listwise)	17					

表 3 标准化后数据集

标准化后数据集			
X_1	X_2	X_3	X_4
−0.076 936	−0.475 61	−0.610 60	−0.631 52
−0.769 36	−0.475 61	−0.610 60	−0.631 52
−0.747 23	−1.053 14	−0.484 01	−0.666 60
−0.739 85	−0.764 37	2.680 69	−0.631 52
−0.710 34	0.390 68	1.541 39	−0.421 01
−0.680 83	0.390 68	−0.610 60	−0.456 10
−0.577 56	−1.341 90	−0.610 60	−0.491 18
−0.511 17	0.101 92	−0.610 60	−0.456 10
−0.474 28	−0.764 37	−0.610 60	−0.561 35
−0.238 23	−0.475 61	−0.230 84	−0.491 18
−0.098 07	−0.764 37	0.528 69	−0.315 76
0.226 51	2.412 03	0.528 69	3.175 13
0.528 96	−0.186 85	−0.610 60	0.333 30
0.728 13	−0.475 61	1.414 81	0.105 25
0.757 64	0.968 21	−0.610 60	−0.035 08
1.008 45	0.968 21	−0.610 60	0.701 69
3.066 58	1.545 73	−0.484 01	1.473 54

下面在标准化后的数据集上做主成分分析，结果见表 4。

表 4 Total Variance Explained

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings		
	Total	% of Variance	Cumulative%	Total	% of Variance	Cumulative%
1	2.365	59.118	59.118	2.365	59.118	59.118
2	1.017	25.434	84.551	1.017	25.434	84.551
3	0.462	11.553	96.104	0.462	11.553	96.104
4	0.156	3.896	100.000	0.156	3.896	100.000

从表 4 中可以看到前三个主成分的累积贡献率为 96.1%，为了达到降维的目的，选取前三个主成分。表 5 是因子载荷矩阵，每一载荷量表示了主成分与对应变量的相关系数，从表中可以更清楚地反映出主成分与各变量的亲疏关系[5]。从结果中可以看出第一主成分与模块规模(X_1)、模块扇出(X_2)、模块控制流路径(X_4)相关性较强，第二主成分与模块扇入(X_3)相关性较强，第三主成分与模块规模(X_1)相关性稍强一点。

表 5 Component Matrix(a)

	Component			
	1	2	3	4
Zscore (X_1)	0.812	−0.125	0.570	0.020
Zscore (X_2)	0.919	0.064	−0.278	0.271
Zscore (X_3)	−0.100	0.987	0.123	0.029
Zscore (X_4)	0.922	0.153	−0.212	−0.285

然后需要求得这三个主成分的表达式。首先要求得特征向量，步骤如下：将前三个因子载荷矩阵输入到数据编辑窗口(为变量 a_1, a_2, a_3)，然后利用“Transform → compute”，输入“$u_1 = a_1$/SQRT(2.365)”，得到特征向量 u_1。同理可得 u_2, u_3。于是主成分表达式为：

$$\begin{cases} y_1 = 0.528x_1 + 0.598x_2 + (-0.065)x_3 + 0.6x_4 \\ y_2 = (-0.124)x_1 + 0.063x_2 + 0.979x_3 + 0.152x_4 \\ y_3 = 0.839x_1 + (-0.409)x_2 + 0.181x_3 + (-0.312)x_4 \end{cases} \quad (2)$$

有了各主成分的表达式，就可以将标准化后的数据代入表达式(2)，计算出各样品的主成分得分，主成分得分见表 6。

表 6 主成分得分

样品编号	第一主成分 y_1	第二主成分 y_2	第三主成分 y_3
1	−1.030	−0.628	−0.364
2	−1.030	−0.628	−0.364
3	−1.393	−0.549	−0.076
4	−1.401	2.572	0.374
5	−0.494	1.558	−0.345
6	−0.360	−0.558	−0.699
7	−1.362	−0.685	0.107
8	−0.443	−0.597	−0.439
9	−1.005	−0.672	−0.021
10	−0.690	−0.301	0.106
11	−0.733	0.434	0.425
12	3.433	1.124	−1.691
13	0.407	−0.624	0.306
14	0.071	1.281	1.029
15	0.998	−0.636	0.140
16	1.572	−0.555	0.121
17	3.459	−0.533	1.393

将这 17 个样品在三维坐标系上描出来，散点图如图 1 所示。

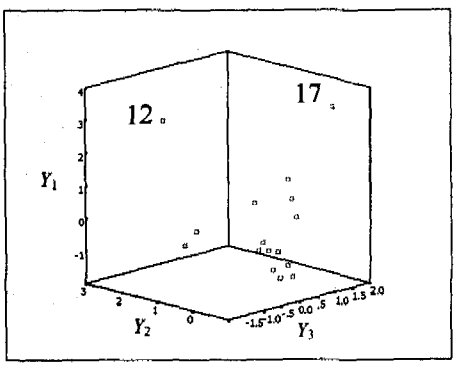

图 1 主成分散点图

从图 1 中可以清楚地看到离群值,离群样本点为样品 12 和 17。关注这两个样品发现,它们的扇出、模块控制流路径值都较大,因此需要对这两个异常样品进行调查分析,从技术因素和人为因素着手,查找引起异常的原因,及时发现潜在问题并作适当调整和改进,以避免在后续开发中带来不必要的损失。图 2 是提取主成分之前,标准化数据集中 X_2(扇出)、X_3(扇入)、X_4(控制流路径)这三个属性的三维三点图,和图 1 相比较,离群值和数据密集部分均不如图 1 明显。在此主成分分析实验中,不仅去除了数据的相关性,而且使数据得到了降维,由四个属性简化为三个主成分,最后由主成分得到的散点图更加清晰地呈现出离群值较大的样品。

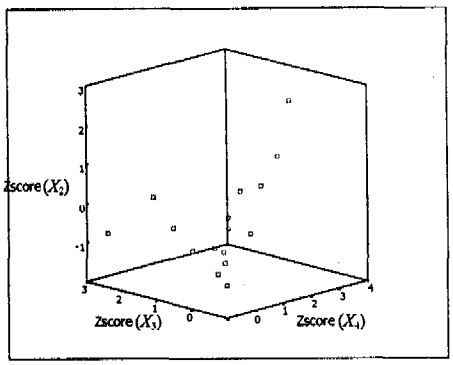

图 2　扇入、扇出及路径的散点图

3　小　结

主成分分析的应用可以减少在软件度量过程中所收集的数据信息在一定程度上的重叠,降低计算量和分析问题的复杂性。因为软件开发中的众多因素之间可能存在一定的相关性,在对某一问题进行研究时很有可能对某些因素的效用进行了重复累加,这样会导致度量结果不能真实地揭示软件开发过程中存在的问题,也就不能进行准确的预测和有效的决策[6]。利用主成分分析,可以找出影响某一软件过程的几个综合指标,使综合指标为原来度量属性的线性组合。综合指标不仅保留了原始度量属性的主要信息,彼此之间又不相关,这样就保证了在软件度量的大量数据中容易抓住问题的主要方面。

参考文献:

[1] Fenton N E, Pfleeger S L. Software Metrics: A Rigorous & Practical Approach[M]. 第 2 版. 北京:清华大学出版社, 2003.

[2] Pillai K, Nair V S S. Statistical analysis of nonstationary software metrics[J]. Information and Software Technology, 1997, 39:363 - 373.

[3] Fenton N E, Neil M. Software metrics: successes, failures and new directions[J]. The Journal of Systems and Software, 1999, 47:149 - 157.

[4] 袁　卫,庞　皓,曾五一. 统计学[M]. 北京:高等教育出版社,2000.

[5] 王　芳. 主成分分析与因子分析的异同比较及应用[J]. 统计教育,2003,56(5):14 - 17.

[6] Maxwell K D, Kusters R J. Software project control and metrics[J]. Information and Software Technology, 2000, 42:963 - 964.

(上接第 143 页)

控制模块包括 256 个指针寄存器,每一个存储器对应一个状态寄存器,指示状态存储器所代表的状态。

从 ACS 单元输入 256 比特的判决信号,控制模块根据这些判决信号,判断每一个前序状态怎样向后序状态转移,形成新的判决信号:不向任何状态转移,用 00 表示;只从上支路转移,用 01 表示;只从下支路转移,用 10 表示;向两条支路同时转移,用 11 表示。

如果一个状态同时向两个状态转移,那么与它同组的状态就不向任何状态转移,这时,通过控制信号把判决比特为 11 的状态寄存器内容复制到判决比特为 00 的状态寄存器中,同时把判决比特一个改为 01,一个改为 10。

最后,根据新的判决信号,为 01 时,状态寄存器和指针寄存器移入 0;为 10 时,状态寄存器和指针寄存器移入 1。

3　结　论

幸存路径存储及输出模块是 Viterbi 译码四大组成模块之一,文中提出的改进的寄存器交换法和传统的寄存器交换法相比,具有占用资源少、存储器访问次数少的特点,降低了功耗,加上寄存器交换法本身速度快,适合在移动通信系统上使用。

参考文献:

[1] 王新梅,肖国镇. 纠错码原理与方法[M]. 西安:西安电子科技大学出版社,2001.

[2] Viterbi A J. Error bounds for convolutional codes and asymptotically optimum decoding algorithm[J]. IEEE Trans Inf Theory, 1967, IT - 13(2):260 - 269.

[3] Forney J G D. The Viterbi algorithm[J]. Proc IEEE, 1973, 61(3):268 - 278.

[4] Kang I, Jr Willson A N. Low - power Viterbi decoder for CDMA mobile terminals[J]. IEEE J. Solid - State Circuits, 1998, 33:473 - 482.

[5] Wicker S B. Error Control Systems for Digital Communication and Storage[M]. Englewood Cliffs, NJ: Prentice - Hall, 1995.