電子情報通信学会論文誌 2002/10 Vol. J85–D–I No. 10

主成分・相関分析によるメトリックスの定量的検証法

阿萬　裕久†（正員）　　山崎　健司†
山田　宏之†（正員）　　野田松太郎†

A Quantitative Method of Verifying Metrics Using Principal Component Analysis and Correlation Analysis

Hirohisa AMAN†, *Regular Member*,
Kenji YAMASAKI†, *Nonmember*,
Hiroyuki YAMADA†, *Regular Member*,
and Matu-Tarow NODA†, *Nonmember*

† 愛媛大学工学部情報工学科，松山市
Department of Computer Science, Faculty of Engineering, Ehime University, Bunkyo-cho 3, Matsuyama-shi, 790–8577 Japan

あらまし　あるソフトウェア属性を測定するためのメトリックスがいくつか与えられたとき，それぞれは異なった視点から独立した測定値を提供しなければならない．換言すると，n 個のメトリックスが与えられていれば，それらによって n 次元の情報が得られなければならない．さもなくば，いくつかのメトリックスが冗長であることになる．そこで本論文では，メトリックスを定量的に検証する一つの方法として，主成分分析・相関分析を用いる方法を提案している．提案手法は，いくつかのメトリックスが与えられたとき，（1）対象としているソフトウェア属性のどれだけの側面を表現できるのか，（2）冗長なメトリックスは含まれていないか，という 2 点について検証を行うことができる．

キーワード　メトリックス，主成分分析，相関分析

1. まえがき

ソフトウェアの設計・開発を支援する一つの有効な方法として，ソフトウェアのもつ様々な属性を評価・予測することが挙げられる．そのための尺度をソフトウェアメトリックス，あるいは単にメトリックスという．メトリックスは，明確な測定目標をもち，その上で客観的かつ正確にソフトウェア属性を測定しなければならない [1]．ソフトウェア属性としては，複雑さ・凝集度・結合度等があるが，一般にそのような属性には様々な側面があり，単一のメトリックスで適切に測定することは難しい [2]．そのため，一つのソフトウェア属性を測定するためにいくつもの異なったメトリックスが考案され，それらを組み合わせることで目的の属性を正確にとらえようとしている．その際，各メトリックスは，それぞれ異なった視点から独立してソフトウェア属性を測定していなければならない．換言すると，あるソフトウェア属性のために n 個のメトリックスが考案されているのであれば，それらによって n 次元の情報が得られなければならない．さもなくば，いくつかのメトリックスが冗長であることになる．

従来，メトリックスが新しく提案されるときは，Weyuker [3] や Briand ら [4] の定性的な検証法（対象とするソフトウェア属性に関する数学的性質を利用）や，開発現場での測定データとの関係（例えば，複雑さメトリックスによる測定値と実際に含まれていたフォールト数との相関 [5] 等）を調べる定量的な検証法が使われてきた．しかしながら，新しく提案されるメトリックスを含めていくつかのメトリックスが与えられたとき，（1）“それらによって対象としているソフトウェア属性のどれだけの側面を表現できるのか，”（2）“冗長なメトリックスは含まれていないか” といった視点での検証は行われていない．新しいメトリックスが提案され，たとえ従来の検証法によってその有効性が示されたとしても，そのメトリックスによる測定が既存のいくつかのメトリックスを合成することで実現されるようではその有意性は低い．そのため本論文では，上述の（1），（2）のような視点での検証を実現するため，主成分分析並びに相関分析 [6] を用いてメトリックスを定量的に検証する方法を提案する．

2. メトリックスの定量的検証法

あるソフトウェア属性に対し，n 個のメトリックス μ_k $(k = 1, \ldots, n)$ が存在しているものとする．このとき，これらのメトリックスの検証法を以下に提案する：

（1） N 個のソフトウェアを無作為に抽出し，各メトリックスによる測定を行う．その結果，各メトリックス値を成分とした n 次元ベクトルが N 個得られる．その各ベクトルを $\boldsymbol{x}_i$ $(i = 1, \ldots, N)$ と表す．

（2） $\boldsymbol{x}_i$ $(i = 1, \ldots, N)$ に対し，各成分間の相関係数を算出し，相関行列 R を求める．

（3） 相関行列 R の固有値を算出する．ここでは，各固有値を λ_j $(j = 1, \ldots, n)$ とする．ただし，$\lambda_1 > \lambda_2 > \cdots > \lambda_n$ である．λ_j に対応する固有ベクトルは第 j 主成分と呼ばれている(注1)．

（4） 各主成分の累積寄与率 α_j $(j = 1, \ldots, n)$ を算出する：

$$\alpha_j = \sum_{t=1}^{j} \lambda_t \Big/ \sum_{t=1}^{n} \lambda_t \ .$$

（注1）：通常，主成分分析では共分散行列の固有値・固有ベクトルを用いるが，各データの単位が異なっている場合はそれらを正規化しなければならない．これは相関行列を用いることを意味する．詳しくは文献 [6] を参照されたい．

ただし，λ_j は相関行列 R の固有値であるため，$\sum_{t=1}^{n}\lambda_t = n$ である．なお，累積寄与率 α_j とは，第 1 主成分から第 j 主成分までの j 個の固有ベクトルを基底とした j 次元空間でもって，もとの n 次元ベクトルの $\alpha_j \times 100$ (%) の情報を表現できることを意味する．

（5） 今，n 次元の情報をより低次元で表現するとしたときに，望ましい累積寄与率の下限を τ とする．すなわち，全データの $\tau \times 100$ (%) の情報を表現できれば十分であるとする．そこで $\alpha_j \geq \tau$ となるような最小の j を求める．もし $j < n$ ならば，測定されたメトリックス値には現実には n 次元の広がりはないことになる．すなわち，n 個のメトリックス μ_k $(k = 1, \ldots, n)$ のうち，いくつかは冗長であると考えることができる．逆に $j = n$ ならば，n 個のメトリックスはいずれを欠くこともできないことがわかる．

なお，τ の値について，本論文では文献 [7] に従い $\tau = 0.95$ とする．

冗長なメトリックスの選定には，各メトリックス間の相関係数（相関行列 R の各成分；対角成分を除く）を用いればよい．つまり，m $(\leq n - j)$ 個のメトリックスを冗長とする場合，まず最も相関の強いメトリックス対を選び，どちらか一方を冗長なメトリックスとみなす．そして，そのメトリックスの存在を無視し，同じ操作を $m - 1$ 回繰り返せばよい．ただし，本論文では相関係数の絶対値が 0.8 以上であるときを “強い相関” とみなし [8]，相関係数の絶対値が 0.8 未満であるようなメトリックス対についてはこの選定の対象外とする．

□

3. 適 用 例

前章で述べた検証法の適用例として，本章ではクラス凝集度メトリックスに対する検証実験の結果を示す．

凝集度は，モジュール内での要素間の機能的な関係付けの強さの度合として定義されている [9]．高い凝集度のモジュールでは，全要素が単一の機能を実現することに関係しており，開発・保守・再利用が容易で，フォールトも含まれにくいといわれている [10]．オブジェクト指向ソフトウェアでは，メソッドや属性をクラスの要素とみなすことで，クラスをモジュールに見立てることができる．つまり，クラスに対しても凝集度の概念を適用できる．

これまでにクラス凝集度メトリックスとしては表 1 に示した八つが知られている．このうち，LCOM1（lack of cohesion in methods-1）及び LCOM2 では，“メソッド同士が同一の属性にアクセスしているか否か” に着目し，そのような組合せの数等を用いることでクラス凝集度を測定している．また，LCOM3 ～ LCOM5 及び TCC（tight class cohesion），LCC（loose class cohesion）では，“メソッド間での属性の共通利用” や “メソッド呼出し” をメソッド間での結合関係ととらえ，そのような関係の広がりや密度等を用いてクラス凝集度を測定している．ICH（information flow-based cohesion）では，メソッド呼出しの回数並びにメソッドのパラメータ数を利用してクラス凝集度を測定している．これらの詳細については文献 [10] を参照されたい．

今回，Sun Java 2 SDK 1.3.1 付属のクラスから 67 個を無作為に抽出し，これら八つのメトリックスを使って測定を行った．その結果に対して主成分分析を施し，その累積寄与率を算出すると表 2 に示す結果が得られた．

累積寄与率は第 5 主成分で 0.95 を超えている．つまり今回の実験データは，現実的には 8 次元の広がりはもっていない．このことから，いくつかのメトリックスは冗長であると推察される．そこで，各メトリックス値間の相関を調べたところ表 3 に示す結果が得ら

表 1 クラス凝集度メトリックス
Table 1 Class cohesion metrics.

メトリックス	説明
LCOM1	属性を共有していないメソッド対の数
LCOM2	LCOM1 −（共有しているメソッド対の数）
LCOM3	属性参照に基づいてグループ化された メソッド集合の数
LCOM4	LCOM3 にメソッド呼出しを加味したもの
LCOM5	メソッド・属性間結合のクラス内での密度
TCC	メソッドによる属性共有の比率
LCC	TCC に間接アクセスを加味したもの
ICH	パラメータ数で重み付けされたメソッド 呼出し回数

表 2 主成分分析の結果：累積寄与率
Table 2 Result of pricipal component analysis: accumulated contribution rate.

主成分	累積寄与率
第 1 主成分	0.508
第 2 主成分	0.733
第 3 主成分	0.845
第 4 主成分	0.945
第 5 主成分	0.977
第 6 主成分	0.995
第 7 主成分	0.999
第 8 主成分	1.000

表 3　各メトリックス間の相関係数
Table 3　Correlation coefficients between each of metrics pairs.

	LCOM1	LCOM2	LCOM3	LCOM4	LCOM5	TCC	LCC	ICH
LCOM1	1	0.999	0.882	0.896	0.0321	−0.0739	−0.118	0.437
LCOM2		1	0.885	0.887	0.0329	−0.0746	−0.119	0.448
LCOM3			1	0.860	0.0676	−0.0752	−0.169	0.692
LCOM4				1	0.0612	−0.161	−0.186	0.316
LCOM5					1	−0.171	−0.283	0.0372
TCC						1	0.727	−0.0145
LCC							1	−0.122
ICH								1

れた．

LCOM1，LCOM2，LCOM3，LCOM4 それぞれの間の相関係数はいずれも 0.8 を上回っており，強い正の相関が確認できる．そのため，これらのうちのいずれか一つと LCOM5，TCC，LCC，ICH の五つのメトリックスを用いて測定を行うことも可能である．このことは，既存のクラス凝集度メトリックスで測定できる情報が現実的には 5 次元であることに対応し，表 2 の累積寄与率とも符合する．なお，今回の測定サンプルの抽出数及び抽出範囲は必ずしも十分ではないため，この結果の意味するところが真に一般的であるかどうかを判定するのは難しい．この問題は，より多くのサンプルをより広範囲にわたって抽出することで解決される．

4. む す び

本論文では，メトリックスを定量的に検証する一つの方法として，主成分分析・相関分析を用いる方法を提案した．本手法は，いくつかのメトリックスが与えられたとき，（1）対象としているソフトウェア属性のどれだけの側面を表現できるのか，（2）冗長なメトリックスは含まれていないか，という 2 点について検証を行うことができる．本手法を従来の検証法と組み合わせることで，より的確なメトリックスの検証が可能になると考えられる．

文　　献

[1] N.E. Fenton and S.L. Pfleeger, Software Metrics: A Rigorous and Practical Approach, 2nd ed., PWS Publishing Company, Boston 1997.

[2] N. Fenton, “Software measurement: a necessary scientific basis,” IEEE Trans. Software Eng., vol.20, no.3, pp.199–206, March 1994.

[3] E.J. Weyuker, “Evaluating software complexity measures,” IEEE Trans. Software Eng., vol.14, no.9, pp.1357–1365, Sept. 1988.

[4] L.C. Briand, S. Morasca, and V.R. Basili, “Property-based software engineering measurement,” IEEE Trans. Software Eng., vol.22, no.1, pp.68–85, Jan. 1996.

[5] V.R. Basili, L.C. Briand, and W.L. Melo, “A validation of object-oriented design metrics as quality indicators,” IEEE Trans. Software Eng., vol.22, no.10, pp.751–761, Oct. 1996.

[6] 塩谷　實，多変量解析概論，朝倉書店，東京，1990.

[7] 本多正久，インフォメーション・アナリストのための多変量解析の実際，産能大学出版部，東京，1993.

[8] 宮本一郎，よくわかる統計学概要，学術図書，東京，1993.

[9] 片山卓也，土居範久，鳥居宏次（監訳），ソフトウェア工学大事典，朝倉書店，東京，1998.

[10] L.C. Briand, J.W. Daly, and J. Wüst, “A unified framework for cohesion measurement in object-oriented systems,” Empirical Software Engineering, vol.3, pp.65–117, 1998.

（平成 14 年 3 月 25 日受付，5 月 15 日再受付）