ソフトウェアプロジェクトデータの分析における課題

門田暁人†

本稿では，ソフトウェアプロジェクトデータの分析者が典型的に直面する課題を整理し，問題提起を行う．

Open Problems in Software Project Data Analysis

Akito Monden†

This paper focuses on typical difficulties in software project data analysis, and proposes topics for workshop discussion.

1. はじめに

筆者はこれまでに，図1のような＜個体×変数＞型のソフトウェアプロジェクトデータを数多く分析してきた．ここでの個体とは，1つの開発プロジェクト，サブシステム，モジュールなどであり，変数はプロダクトメトリクス，プロセスメトリクス，資源メトリクスを含む．本稿では，筆者の経験を踏まえ，この種のデータ分析における課題について問題提起したい．

図 1 に類似するソフトウェアプロジェクトデータは，多くのソフトウェアベンダーで収集され，品質保証部門や生産管理部門で活用されている．一般に知られているデータセットとしては，International Software Benchmarking Standards Group (ISBSG)が 20 カ国から収集した約3000プロジェクト， 99変数の実績データ[2]，情報処理推進機構（IPA）ソフトウェアエンジニアリングセンターが日本のソフトウェアベンダー15 社から収集した 1009 プロジェクト，約 400 変数の実績データ[1]，NASA において収集された10プロジェクト，約 14000 モジュール，27 変数 の実績データ[3]などがある．

これらの開発実績データは，人的要因が介在するために，(1)各変数の定義が非厳密である，(2)計測の信頼性や計測方法にばらつきがある，(3)欠損値が存在する，といった問題があり，統計的な分析を行うに先立って，データの前処理や取捨選択が必要となる．また，この種のデータの特徴として，(1)量的データと質的データが混在している，(2)1 つの個体（プロジェクト）における各変数の計測時期に順序関係が存在する，(3)量的変数は値の大きな偏りがあり，正規分布とならない，といった事情があり，統計的分析を困難にしている．これらの事情を踏まえて，以降では，ソフトウェアプロジェクトデータの分析において留意すべき課題を述べる．

プロジェクトID	開発種別	業種	アーキテクチャ
1	a: 新規開発	a: 銀行	a: クライアントサーバ
2	a: 新規開発		b: スタンドアロン
3	a: 新規開発	a: 銀行	b: スタンドアロン
4		a: 銀行	b: スタンドアロン
5	b: 改修・保守	b: 製造業	c: 混合
6	a: 新規開発	a: 銀行	b: スタンドアロン
7	b: 改修・保守	a: 銀行	b: スタンドアロン
8	a: 新規開発	c: 公共	c: 混合

要求仕様_明確度合	開発期間(月数)	規模(FP)	開発工数(人時)
	15	556	24690
c: ややあいまい	8	80	825
		77	758
a: 非常に明確	4	255	2119
	6	349	2741
d: 非常にあいまい	1		1090
b: かなり明確	4	375	1855
	6	271	1747

Fig 1. ＜個体×変数＞型データの例

2. データ分析に先立つ課題

2.1. 変数の定義の厳密さ

ソフトウェアプロジェクトデータにおいて，各変数の定義が厳密でないことがしばしばある．例えば，「要求仕様の明確度合」という変数（順序尺度）の取りうる値として，「非常に明確」「かなり明確」「ややあいまい」「非常にあいまい」の4つのカテゴリを考えた場合，各カテゴリの境界を厳密に定めることは難しい．データ分析者は，各変数の定義の厳密さを念頭において分析を進める必要がある．

2.2. 計測方法のばらつき

変数の定義が厳密であっても，定義どおりに計測で

†奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology

きるとは限らない．例えば，「ソースコード行数」の定義にあたっては，空行やコメント行の取り扱い，ツールにより自動生成された行の取り扱い，既存システムの流用・改造時の行の取り扱いなどをプログラミング言語ごとに決定する必要があるが，細かい取り決めをすればするほど，定義どおりの計測が困難となり，計測方法がばらつく恐れがある．

別の例として，GNATS や Gugzilla をはじめとする多くの障害管理ツールでは，「障害の重要度」の入力項目があるが，デフォルト値が 3（重要度＝中）であるために 3 がそのまま入力され，実態を反映したデータが得られないことがある．

このような現状から，分析者は，各変数の定義の厳密さを精査するのみならず，定義通りの計測が行われているかについても注意を払う必要がある．

2.3. 欠損値

ソフトウェアプロジェクトデータは，一般に，多数の欠損値を含む．一方，多くの統計手法は，欠損値がないことを前提としている．そのため，何らかの方法で予め欠損値を補ったり，欠損を含む個体や変数を除去する必要がある．例えば，名義尺度の場合は，「不明」といったカテゴリを新たに設けて欠損値を割り当てることが考えられる．順序尺度の場合は，欠損値を中央値（5 段階の順序であれば 3）に割り当てることが考えられる．量的データの場合は，欠損値を当該変数の平均値や中央値で埋めたり，相関の高い他の変数を用いた単回帰分析により値を推定することが考えられる．

ただし，このように欠損値を埋めることは，欠損率の低い変数に限ることが望ましい．欠損率の高い個体や変数については，値を埋めることを考えず，データセットから除去することも場合によっては必要である．また，欠損値を埋めた場合は，分析結果への影響を考慮する必要がある．

なお，変数ごとの基礎統計量（平均値，中央値，分散など）の算出や，散布図，相関係数，アソシエーション分析などにおいては，欠損値を埋めずに行うことが望ましい．

3. データ分析時の課題

3.1. 量的データと質的データの混在

プロジェクトデータは量的データ（比尺度，間隔尺度）と質的データ（順序尺度，名義尺度）の両方を含むが，多くの統計手法は，量的データと質的データを同時に取り扱うことができないため，工夫が必要となる．

量的データである比尺度と間隔尺度については，適用すべき統計手法に決定的な違いはない．また，これらの尺度は順序尺度や名義尺度への変換が可能である．一方，順序尺度は，実用上は間隔尺度とみなして線形回帰分析等の説明変数に用いることもあるが，本来は許容されない．名義尺度については，複数の 2 値データに変換することで，量的データとして扱う場合もある．

3.2. 変数の計測時期

「開発種別」や「業種」といった変数は，システム化計画の時点で決定されることが多いが，アーキテクチャは要求分析の段階，プログラミング言語は基本設計の段階にならないと決まらないことが多い．何らかの予測モデルを構築する場合，各変数の値が決定される時期を考慮して説明変数と目的変数を決める必要がある．例えば，開発種別，業種，規模（FP）から開発総工数を予測することは問題ないが，開発総工数から規模(FP)を予測することは意味がない．

3.3. 値の分布

多くの量的変数は，正規分布とならない．ところが，多くの統計手法は，標本の母集団が正規分布であることを仮定しているため，注意が必要である．一つの解決方法として，値の対数変換を行ったり，正規分布を仮定しない手法（ノンパラメトリック検定など）を用いることが考えられる．

4. おわりに

本稿では，ソフトウェアプロジェクトデータの分析者が典型的に直面する問題に着目し，分析に先立つ課題と分析時の課題に分けて整理した．ワークショップでは，ソフトウェア開発データの実態に即した分析はどうあるべきかを議論したい．

参考文献

[1] 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター: ソフトウェア開発データ白書2005 ～IT 企業 1000 プロジェクトの定量データを徹底分析～，日経 BP 社， 2005.

[2] International Software Benchmarking Standards Group: ISBSG Estimating, Benchmarking and Research Suite Release 9, 2004, http://www.isbsg.org/

[3] NASA IV&V Facility Metrics Data Program, http://mdp.ivv.nasa.gov/