類似性に基づく工数見積手法への偏差パターン類似度の適用

柿元 健[1]　池内 伸多郎[1,†1]

概要：本稿では，類似性に基づく工数見積手法のプロジェクト間の類似度算出に偏差パターン類似度を適用した結果について報告する．偏差パターン類似度は，プロジェクト規模を考慮しており，欠損値を含むデータに適用可能な類似度である．

Applying Deviation Pattern Similarity for Estimation by Analogy

1. はじめに

ソフトウェア開発において，予算の見積は納期や要員数の算出につながるため，プロジェクトを開始する上で必須事項である．ソフトウェア開発における予算の大半は人件費が大部分を占めるため，一般的に工数を見積もることによって決定する．

工数見積において，これまでに様々な手法が提案されてきているが，本稿では，現行プロジェクトに類似している過去プロジェクトの工数に基づいて工数を見積もる手法である類似性に基づく見積手法（Estimation by Analogy 手法　以下，EbA 手法）[4][5] を扱う．EbA 手法の類似度算出法として，ユークリッド距離，あるいは，コサイン類似度が一般的に用いられている．しかし，ユークリッド距離は欠損値を含むデータに対してそのまま適用できず，コサイン類似度はプロジェクト規模を考慮できないという弱点がある．

そこで，本稿では，欠損値を含むデータにも適用可能で，プロジェクト規模も考慮できる類似度算出法である偏差パターン類似度を EbA 手法に用いた結果について報告する．

2. 類似性に基づく工数見積手法

類似性に基づく見積手法（EbA 法）は，工数が類似しているプロジェクト同士はプロジェクトの内容も類似しているという仮定に基づき，類似しているプロジェクトの工数を基に見積を行う手法である．

EbA 手法の手順と本稿で用いる手法は以下のとおりである．

- ダミー変数化
  カテゴリ変数のメトリクスが含まれる場合，量的データとして扱うためカテゴリごとのダミー変数に置き換える．
- 正規化
  メトリクス毎に値域を揃え，各メトリクスの類似度への影響を均等にするために正規化を行う．最小値 0，最大値 1 とする正規化を用いる．
- 類似度算出
  正規化したメトリクスの値を用いて見積プロジェクトと過去プロジェクトの類似度を算出する．類似度算出にはコサイン類似度，ユークリッド距離に基づいた類似度，偏差パターン類似度を用いる．
- 見積値算出
  類似度が高い過去プロジェクトの実績工数から，見積対象プロジェクトの見積工数を算出する．見積値算出には類似度を重みとした加重平均を用いる．

3. 偏差パターン類似度

偏差パターン類似度は，平均値との偏差を要素とした偏差ベクトル間のコサイン類似度であり，平均値を基準としてどの程度類似しているかを示す．以下にコサイン類似度の算出方法を示す．

(1) メトリクスごとに平均値を算出し，各平均値を要素とする平均ベクトルを算出する．欠損値は平均値の算出には含めない．

[1] 香川高等専門学校電気情報工学科
National Institute of Technology, Kagawa College, Takamatsu, Kagawa 768–8058, Japan
[†1] 現在，(株) ドワンゴ
Presently with DWANGO Co., Ltd.

(2) 各プロジェクトにおいて各メトリクスの値とそのメトリクスの平均値との差を要素とした偏差ベクトルを算出する．各プロジェクトをひとつのベクトルとみなした場合，プロジェクトのベクトルから平均ベクトルを引いたベクトルが偏差ベクトルとなる．欠損値が含まれる場合，欠損値のメトリクスの要素は偏差ベクトルには含まれない．

(3) 偏差パターン類似度として，偏差ベクトル間のコサイン類似度を算出する．どちらかのプロジェクトに欠損値が含まれる場合，そのメトリクスは算出には使用しない．

なお，メトリクス間の偏差パターン類似度を求めた値が相関係数となる．

4. 評価実験

本稿では，EbA 手法において，類似度算出に偏差パターン類似度を用いた場合の見積精度と，既存の類似度計算であるユークリッド距離とコサイン類似度を用いた場合の見積精度の比較を行った．

評価実験は，欠損値を含まないデータセットと欠損値を含むデータセットに対して行った．欠損値を含まないデータセットとして，tera-PROMISE リポジトリ [3] の Effort に属する albrecht，kemerer，miyazaki94 の 3 種類のデータセットを用いた．欠損値を含むデータセットには，ISBSG が収集したデータ [1] を基に作成した，プロジェクト数 1857 件，メトリクス数 12 個，データ欠損率 52.98%のデータセットを用いた．

Analogy 数は，欠損値を含まないデータセットでは 1〜データセットのプロジェクト数 −1 まで変化させ，欠損値を含むデータセットでは 1〜3 で変化させた．

評価実験の方法として交差検証法のひとつである Leave-one-out 法を用い，評価尺度には過大見積，過小見積ともに評価可能な MBRE(Mean Balanced Relative Error)[2] を用いた．MBRE は式 (1) で表わされる．

$$MBRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|X_i - E_i|}{\min(X_i, E_i)} \quad (1)$$

ここで，n はプロジェクト数，X_i はプロジェクト i の実測値，E_i はプロジェクト i の見積値を表す．

5. 結果と考察

実験の結果を表 1 に示す．紙面の都合上 Analogy 数が 1〜3 の結果についてのみ記載している．

欠損値を含まないデータ（albrecht, kemerer, miyazaki94）においては，多くの場合，ユークリッド距離に基づいた類似度が最も見積精度が高く，次に偏差パターン類似度がユークリッド距離に基づいた類似度よりも少し精度が低く，コサイン類似度が最も見積精度が低くなった．

表 1 各手法の Analogy 数 1〜3 の MBRE

データセット	類似度	Analogy 数		
		1	2	3
albrecht	コサイン類似度	3.40	2.15	2.21
	ユークリッド距離	1.11	0.97	0.97
	偏差パターン類似度	0.67	1.02	1.15
kemerer	コサイン類似度	1.48	1.04	1.16
	ユークリッド距離	1.12	0.95	0.92
	偏差パターン類似度	1.20	1.01	1.05
miyazaki94	コサイン類似度	2.21	2.22	2.22
	ユークリッド距離	1.33	1.06	1.01
	偏差パターン類似度	1.74	1.29	1.22
ISBSG	コサイン類似度	3.29	2.48	2.29
	ユークリッド距離	5.04	3.91	3.68
	偏差パターン類似度	3.18	2.24	2.22

一方，欠損値を含むデータセット（ISBSG）においては，偏差パターン類似度が最も見積精度が高く，次にコサイン類似度が偏差パターン類似度よりも少し精度が低く，ユークリッド距離に基づいた類似度が最も見積精度が低くなった．

以上の結果より，Analogy 数を増やしていっても，偏差パターン類似度は，欠損値を含まないデータセットにおいては，ユークリッド距離に基づいた類似度よりも少し精度は低いが同様の傾向を示した．また，欠損値を含むデータセットにおいては，コサイン類似度よりも少し精度が高く同様の傾向を示しているといえる．従って，ユークリッド距離の欠損値を含むデータに対してそのまま適用できない，コサイン類似度のプロジェクト規模を考慮できないという弱点を克服できていると考えられる．

6. おわりに

本稿では，類似性に基づく工数見積手法の類似度計算において偏差パターン類似度を適用した結果について示した．ワークショップでは，偏差パターン類似度をより効果的に適用するための方法や適用分野について議論したい．

参考文献

[1] ISBSG Estimating：Benchmarking and Research Suite Release 11: International Software Benchmarking Standards Group, (2009), http://www.isbsg.org/

[2] Kitchenham, B.A., Pickard, L.M., MacDonell, S.G., Shepperd, M.J., :What Accuracy Statistics Really Measure, IEEE Software, Vol.148, No.3 ,pp.81–85, (2001).

[3] Menzies, T., Krishna, R., Pryor, T., :The Promise Repository of Empirical Software Engineering Data, http://openscience.us/repo, (2016).

[4] Shepperd, M., Schofield, C., :Estimating software project effort using analogies, IEEE Transaction Software Engineering, Vol.23, No.12, pp.736–743, (1997).

[5] 角田雅照, 大杉直樹, 門田暁人, 松本健一, 佐藤慎一, :協調フィルタリングを用いたソフトウェア開発工数予測方法, 情報処理学会論文誌, Vol.46, No.5, pp.1156–1164, (2005).