

ソフトウェア開発工数予測ツールの開発

非会員 東京都立大学 *酒井日向 SAKAI Hyuuga
05000041 東京都立大学 肖霄 XIAO Xiao

1. はじめに

ソフトウェア開発プロジェクト管理において、計画段階でソフトウェア開発工数(人月)を精度よく予測しておくことは、コストの超過や納期の遅れなどを未然に防ぐことができるため、プロジェクトを成功させる上で重要である。

ソフトウェア開発工数を予測する手法は多種多様に存在する中で、線形重回帰に基づく手法は多くの利点を有するものの、説明変数の間に従属性や多重共線性がみられると、効率良く変数選択ができないなどの問題点がある。これに対して、罰則項付回帰は正則化項を取り入れることによって、目的変数に影響のある(あるいは強い)説明変数だけ自動的に選択でき、工数予測に効果的であることが確認されている [1]。

他方、ソフトウェア信頼性向上および予測に関する研究の学術的成果を実装し、ソフトウェア開発支援ツールとして提供する研究プロジェクトも多数存在する [2][3][4]。しかし、これらはソフトウェア不具合の発生メカニズムに焦点をあてたものであり、ソフトウェア開発工数予測を目的としたツールは未だにないのが現状である。よって、本研究ではソフトウェア開発工数予測に特化した予測ツールの開発を目指す。

2. 実装する工数予測手法

肖ら [1] は類推手法と罰則項付回帰を融合し、類似性尺度により選出した類似プロジェクトのみを用いた罰則項付回帰による工数予測手法を提案し、従来手法より予測精度が高くなることを確認した。本研究では、肖ら [1] の罰則項付ハイブリッド回帰を基に実装を行う。

線形重回帰モデルは、

$$Y = \sum_{j=0}^n \beta_j X_j + \varepsilon \quad (1)$$

のように表される。ここで、 Y はプロジェクトの開発工数、 X_j ($j = 1, \dots, n$) はプロジェクトの特性 j 、 β_j ($j = 0, \dots, n$) は回帰係数、 ε は確率誤差項である。ここで、一般性を失うことなく、 $X_0 = 1$ と仮定する。

過去の全プロジェクトを入力データとして与える代わりに、類似プロジェクトのみを用いると観測データに含まれる外れ値などの影響を取り除き、予測精度が向

上する。新規プロジェクトと過去プロジェクトの類似性尺度を計算し、最も類似する過去プロジェクトの数が k 個とすると、類似性に基づいた線形重回帰モデルの回帰係数を最小二乗 (Ordinary Least Squares; OLS) 法により求めた OLS 推定値は式 (2) によって与えられる。

$$\hat{\beta}_{OLS}^{ana} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^k (y_i - \sum_{j=0}^n \beta_j x_{i,j})^2 \quad (2)$$

ここで、 β ($= \beta_0, \dots, \beta_n$) は回帰係数、 y_i と $x_{i,j}$ はそれぞれプロジェクト i の開発工数と特性 j を表す。OLS 法の目的関数に罰則項を加えると、Ridge 推定値と Lasso 推定値はそれぞれ式 (3)、式 (4) によって与えられる。

$$\hat{\beta}_{Ridge}^{ana} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^k (y_i - \sum_{j=0}^n \beta_j x_{i,j})^2 + s \sum_{j=1}^n \beta_j^2 \right\} \quad (3)$$

$$\hat{\beta}_{Lasso}^{ana} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^k (y_i - \sum_{j=0}^n \beta_j x_{i,j})^2 + s \sum_{j=1}^n |\beta_j| \right\} \quad (4)$$

ここで、 $s \sum_{j=1}^n \beta_j^2$ と $s \sum_{j=1}^n |\beta_j|$ は Ridge 罰則項と Lasso 罰則項である。また、 s (≥ 0) は β に課する縮小量をコントロールするパラメータであり、一般化クロスバリデーション法により決定する。 $s = 0$ のときは $\hat{\beta}_{Ridge} = \hat{\beta}_{Lasso} = \hat{\beta}_{OLS}$ となり、 s が大きくなると罰則が強くなる。

3. システム概要

本研究で開発するツールのシステム構成図を図 1 に示す。点線部分と実線部分はそれぞれ、フロントエンドとバックエンドを表しており、PHP と R を用いてそれぞれ実装する。

フロントエンドでは、ユーザーに過去プロジェクトのデータ(開発工数と特性)と新規プロジェクトのデータ(特性のみ)をアップロードしてもらい、予測結果のレポートを Excel ファイル形式でユーザーに返す。詳細は後述するが、レポートには、最良手法とそれによる新規プロジェクトの開発工数の予測値に加え、参考情報として前処理の結果、新規プロジェクトの類似プロジェクトデータセット、12 通りの予測手法の予測精度(SA 値)などを記録する。

バックエンドではまずデータセットの前処理を行う。詳細は割愛するが、エラーチェック、従属変数の削除、および説明変数間の多重共線性の有無を確認する。次にユーザーデータセットにとって最良な工数予測手法を特定する。類似性尺度はユークリッド距離 (Euclidean Distance; ED) を、類似プロジェクトの個数の決定は四分位数法 [1] を採用する。四分位数は $l = 1, 2, 3, 4$ の 4 通りがあり、線形重回帰モデルの推定法は、OLS 推定, Ridge 推定, Lasso 推定の 3 通りがあるため、工数予測手法は 12 通りある。具体的には、ユーザーデータセットの過去プロジェクトの中から 1 つ選択し、仮新規プロジェクトとする。仮新規プロジェクトとそれ以外の過去プロジェクトとの ED を求め、小さい順に並べ替えた第 l 四分位数にあたる ED を Q_l とする。仮新規プロジェクトとの ED が Q_l よりも小さい過去プロジェクトを類似プロジェクトとして選出する。仮新規プロジェクトの類似プロジェクトデータセット DS_{Q_l} ($l = 1, 2, 3, 4$) ごとに 3 通りの推定法を適用し、12 通りの開発工数の予測値を求めることができる。これをユーザーデータセットの過去プロジェクトの全て (m 件) に対して行うと、開発工数の予測値 ${}^q\hat{g}_i^r$ ($i = 1, 2, \dots, m$, $r \in \{OLS, Ridge, Lasso\}$, $q \in \{Q_1, Q_2, Q_3, Q_4\}$) が得られる。これらの予測値を基に、SA を評価基準として 12 通りの予測手法の予測精度を評価し、最も大きい評価値を示す手法をユーザーデータセットにとって最良な工数予測手法として定める。

$$SA = \left(1 - \frac{MAE}{MAE_{p0}}\right) \times 100 \quad (5)$$

ここで、 $MAE = \frac{1}{m} \sum_{i=1}^m |y_i - {}^q\hat{g}_i^r|$, $MAE_{p0} = \frac{2}{m^2} \sum_{i=1}^m \sum_{j=1}^{j<i} |y_i - y_j|$ である。

最後に、新規プロジェクトの工数予測を行う。仮に Ridge 推定と Q_2 四分位数の組合せが最良な工数予測手法として定められたとする。新規プロジェクトとの ED が Q_2 よりも小さい過去プロジェクトを類似プロジェクトとして選出し、Ridge 推定を用いて開発工数の予測値を算出する。

4. まとめと今後の課題

本研究ではソフトウェア開発工数予測ツールの開発を行った。現段階では、推定法に OLS 推定, Ridge 推定, Lasso 推定, 類似性尺度に ED, 類似プロジェクトの決定方法に四分位数法を用いた。今後は、Elastic Net, SCAD, MCP など他の罰則項付推定, 重み付きユークリッド距離や余弦類似度など他の類似性尺度, fixed-k 法やクラスタリングなど他の類似プロジェクトの決定方法を追加する予定である。

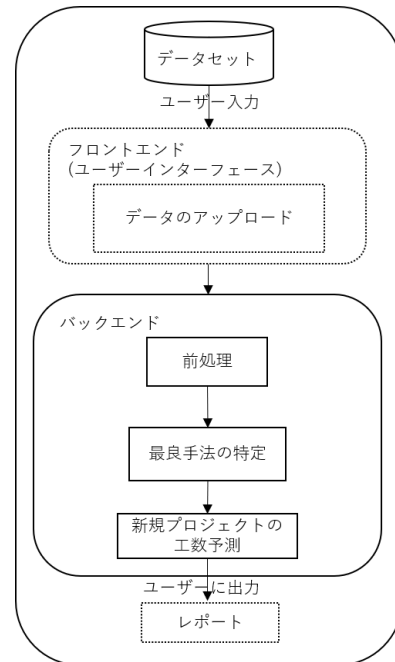


図 1: システム構成図

また、今回は専門的知識をほとんど持たない一般ユーザーを想定しており、ユーザー側のアクションをデータセットのアップロードのみに限定したが、今後は、推定法や類似プロジェクトの決定方法などを選択できるようにし、より幅広いユーザー層を視野に入れて改良する必要がある。さらに、R パッケージとして CRAN に投稿する予定である。

参考文献

- [1] 肖霄, 土肥正, “プロジェクト類似性に基づいた線形重回帰モデルによるソフトウェア開発労力の予測,” 情報処理学会 SES2013 論文集, pp. 1-8, 2013.
- [2] H. Okamura and T. Dohi, “SRATS: Software reliability assessment tool on spreadsheet (Experience report),” Proceedings of 2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE2013), pp. 100–107, 2013.
- [3] J. Wu, T. Dohi and H. Okamura, “W-SRAT: Wavelet-based Software Reliability Assessment Tool,” Proceedings of 2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS2021), pp. 564–573, 2021.
- [4] 水野 修, 黒田 翔太, 石原 一宏, 山下 大輔, “テキスト分類による不具合予測システムの実装と企業環境での評価”, 信学技報, vol. 120, no. 343, SS2020-17, pp. 19-24, 2021.