

NAIST-IS-DD0661016

博士論文

プロジェクトの個別性を考慮した
定量的ソフトウェア開発工数見積り

戸田 航史

2009年 11月 5日

奈良先端科学技術大学院大学
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
博士(工学) 授与の要件として提出した博士論文である。

戸田 航史

審査委員：

松本 健一 教授 (主指導教員)

関 浩之 教授 (副指導教員)

飯田 元 教授 (副指導教員)

門田 暁人 准教授 (副指導教員)

プロジェクトの個別性を考慮した 定量的ソフトウェア開発工数見積り*

戸田 航史

内容梗概

ソフトウェア開発の初期段階において、開発に必要な労力（開発工数）を高い精度で見積ることが必須である。ただし、ソフトウェア開発は個別性が高いために、あらゆる開発プロジェクトに適合する万能の見積り方法は存在しない。本論文では、従来広く用いられてきた工数見積り方法である線形重回帰モデルによる見積り方法とアナロジーベース法それぞれに対し、開発プロジェクトの個別性を考慮した拡張を行った。これにより平均的な見積り精度の向上と見積りを大きく外すプロジェクトの低減を実現できた。提案する拡張方法の概略は次の通りである。

(1) プロジェクトの特性に応じたフィットデータの選定方法

提案する方法では、重回帰モデルに基づく工数見積り方法に対し、モデル構築のためのプロジェクトの集合（フィットデータ）を選定する手順を追加する。見積り対象プロジェクトの特性に基づいてフィットデータを選定することにより、個々のプロジェクト向けにカスタマイズされた重回帰モデルの構築を行う。

ソフトウェア工学における代表的な実験用データセットである ISBSG データセットを用いた評価実験の結果、フィットデータの選定を行わない従来の重回帰モデルと比較すると、分母を実測値とする相対誤差 MRE (Magnitude of Relative Error) の中央値が 0.552 から 0.383 へ、分母を予測値とする相対誤差 MER (Magnitude of Error Relative) の中央値が 0.457 から 0.381 となり、見積り精度が向上した。

(2) 類似プロジェクトの特性に応じた見積り回避プロジェクトの選定方法

* 奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 博士論文, NAIST-IS-DD0661016, 2009 年 11 月 5 日.

アナロジーベース法は，開発プロジェクトの個別性を考慮した見積り方法であり，見積り対象の開発プロジェクトに対し，類似するプロジェクト群を過去の実績データから選定し，それらの実績工数に基づいて開発工数を見積る方法である．ただし，プロジェクトによっては，極端に見積り精度が低くなる（見積りに失敗する）ことが課題であった．

提案する方法では，アナロジーベース法を拡張し，選定された類似プロジェクト群の特徴に基づいて，見積り精度が低くなるか否かを見積り，見積りを実施すべきか否かを判断する．見積りを実施すべきでないと判断されたプロジェクトは重回帰モデルによる見積りを行い，そうでないプロジェクトでは従来通りアナロジーベース法による見積りを行う．ISBSG データセットを用い，類似度に Euclidian Similarity を用いた評価実験の結果，常にアナロジーベース法を用いた見積りを行う場合と比べて，MRE の相対誤差が 100%を超えた（見積りに失敗した）プロジェクトの割合である $\text{Pred}(\overline{100})$ が最大で 0.254，平均で 0.198 減少し，見積りを大きく外すプロジェクトが減少した．

キーワード

工数見積り, 重回帰分析, モデル選択, プロジェクト件数, 見積りの失敗

Customizing Effort Estimation for an Individual Software Development*

Koji Toda

Abstract

Accurate effort estimation is one of the most important requirements in an early stage of software development. However, there is no universal method or model that can estimate various types of projects' effort with high accuracy since each software project has its own characteristics or factors that affect the effort. Therefore, we need to take each project's characteristics into account when building an estimate. To build a custom-made estimate to an individual project, this paper proposes two methods that use historical project data;(1) extension of the multivariate regression model and (2) extension of the analogy-based method. The main achievements are as follows.

(1) A fit data selection method using target project features:

This paper extends conventional multivariate regression modeling by adding a procedure to select fit data, which is a set of past projects for model construction. The proposed method enables us to build a customized regression model for an individual project by selecting the fit data based on the estimation target project's features. We conducted an experiment to evaluate the proposed method using the ISBSG data set, which is the most commonly used project data set in software engineering researches. Compared to conventional regression modeling, the MdMRE was improved from 0.552 to 0.383 and the MdMER improved from 0.457 to 0.381.

* Doctoral Dissertation, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD0661016, November 5, 2009.

(2) A method for identifying failed estimation based on features of neighborhood projects:

The analogy-based method can build a customized estimation for an individual project since it selects neighborhood projects of the target project from a historical dataset and estimates target project's effort from neighborhood projects' effort. However, the problem with the analogy-based method is that estimation accuracy becomes extremely bad for some projects. The proposed method identifies such projects that are likely to fail in estimation based on features of neighborhood projects. The identified projects are estimated by the regression model instead of the analogy-based method. As a result of an experimental evaluation using the ISBSG dataset, $\text{Pred}(\overline{100})$ was improved 0.254 in maximum and 0.198 in average.

Keywords:

effort estimation, multi-regression analysis, model selection, number of projects, estimation failure

関連発表論文

学術論文誌

1. 戸田 航史, 門田 暁人, 松本 健一, ソフトウェア開発工数予測のためのフィットデータ選定方法, 情報処理学会論文誌 (採録決定). (第3章に関連する)

国際会議発表

1. Koji Toda, Akito Monden, and Ken-ichi Matsumoto, Fit data selection based on project features for software effort estimation models, IASTED International Conference on Advances in Computer Science and Engineering (IASTED ACSE 2010) (submitted). (第3章に関連する)
2. Koji Toda, Akito Monden, and Ken-ichi Matsumoto, Fit data selection for software effort estimation models, In Proceedings of the 2nd International Symposium on Empirical Software Engineering and Measurement (ESEM'08), pp.360-361, Kaiserslautern, Germany, October 2008. (第3章に関連する)

査読付き国内研究集会発表

1. 戸田 航史, 門田 暁人, 松本 健一, ソフトウェア開発工数予測における予測回避プロジェクトの選定とその効果, ソフトウェア工学の基礎 XVI, 日本ソフトウェア科学会 FOSE2009 (採録決定). (第4章に関連する)

国内研究集会発表

1. 戸田 航史, 工数見積もりモデルの予測精度を低下させるプロジェクトの特徴分析, 情報処理学会, ウィンターワークショップ 2006・イン・鴨川論文集, Vol.2006, No.2, pp.71-72, January 2006. (第2章に関連する)
2. 戸田 航史, 角田 雅照, 門田 暁人, 松本健一, 工数見積もりモデルで予測で

きないソフトウェアプロジェクトの特徴分析, 電子情報通信学会技術報告,
ソフトウェアサイエンス研究会, No.SS2005-74, pp.67-72, December 2005.
(第2章に関連する)

その他の発表論文

学術論文誌

1. 田村 晃一, 柿元 健, 戸田 航史, 角田 雅照, 門田 暁人, 松本 健一, 大杉 直樹, 工数予測における類似性に基づく欠損値補完法の実験的評価, コンピュータソフトウェア, Vol.26, No.3, pp.44-55, August 2009.
2. 瀧 進也, 戸田 航史, 門田 暁人, 柿元 健, 角田 雅照, 大杉 直樹, 松本 健一, プロジェクト類似性に基づく工数見積もりに適した変数選択法, 情報処理学会論文誌, Vol.49, No.7, pp.2338-2348, July 2008.

国際会議発表

1. Koichi Tamura, Takeshi Kakimoto, Koji Toda, Masateru Tsunoda, Akito Monden, and Ken-ichi Matsumoto, Empirical evaluation of missing data techniques for effort estimation, In Proceedings of International Workshop on Software Productivity Analysis And Cost Estimation (SPACE2008), pp.4-9, Beijing, China, December 2008.

査読付き国内研究集会発表

1. 田村 晃一, 柿元 健, 戸田 航史, 角田 雅照, 門田 暁人, 松本 健一, 大杉 直樹, 工数予測における類似性に基づく欠損値補完の効果, ソフトウェア工学の基礎 XIV, 日本ソフトウェア科学会 FOSE2007, pp.85-93, November 2007.

国内研究集会発表

1. 渡邊 瑞穂, 柿元 健, 戸田 航史, 門田 暁人, 松本 健一, ソフトウェア開発工数予測における特異プロジェクト除去の効果, 日本ソフトウェア科学会第24回大会論文集 CD-ROM(講演番号 7B-2), September 2007.
2. 田村 晃一, 柿元 健, 戸田 航史, 角田 雅照, 門田 暁人, 松本 健一, プロジェク

ト間の類似性に基づくソフトウェアメトリクスの欠損値の補完, ソフトウェア信頼性研究会 第4回ワークショップ, pp.17-23, June 2007.

3. 戸田 航史, 中道 上, 島 和之, 大平 雅雄, 阪井 誠, 松本 健一, Web ページ閲覧者の視線に基づいた情報探索モデルの提案, 情報処理学会研究報告, HI 研究会, Vol.2005-HI-113, No.52, pp.35-42, May 2005.

目次

第1章	はじめに	1
1.	研究の背景と概要	1
2.	論文構成	4
第2章	特性を考慮した工数見積りモデルの性能調査	5
1.	まえがき	5
2.	ステップワイズ重回帰分析	7
3.	ISBSG データセット	8
4.	工数見積りモデルの精度を低下させる要因の特定	11
4.1	特定のための実データを用いた実験	11
4.1.1	概要	11
4.1.2	手順	12
4.1.3	結果	13
4.1.4	考察	16
5.	プロジェクト特性による層別モデルと非層別モデルの見積り性能の比較	17
5.1	性能比較実験	17
5.1.1	概要	17
5.1.2	手順	18
5.1.3	結果	18
5.1.4	考察	19
6.	関連研究	22
7.	まとめ	22

第3章	ソフトウェア開発工数見積りのためのフィットデータ選定方法	24
1.	まえがき	24
2.	重回帰モデル構築におけるフィットデータ選定の問題	26
3.	提案するフィットデータ選定方法	28
4.	ケーススタディ	30
4.1	概要	30
4.2	アナロジーベース法による工数見積り	32
4.3	利用したデータセット	34
4.4	精度評価指標	35
4.5	モデル適合度と精度の関係の調査	36
4.5.1	調査概要	36
4.5.2	調査手順	37
4.5.3	調査結果	37
4.6	結果	40
4.7	考察	43
5.	まとめ	45
第4章	ソフトウェア開発工数見積りにおける見積り回避プロジェクトの選定とその効果	47
1.	まえがき	47
2.	仮説	48
3.	ケーススタディ	51
3.1	概要	51
3.2	利用したデータセット	52
3.3	精度評価指標	53
3.4	手順	54
3.5	結果	55
3.5.1	ばらつきの指標と MRE の相関係数	55
3.5.2	ばらつきと $\text{Pred}(\overline{100})$ の関係	57
3.6	考察	64

4. まとめ	67
第5章 アナロジーベース法と対数重回帰モデルを組み合わせた工数見積り 方法の提案	68
1. まえがき	68
2. ケーススタディ	69
2.1 概要	69
2.2 手順	69
2.3 結果	70
2.4 考察	71
3. まとめ	77
第6章 おわりに	79
謝辞	81
参考文献	84
付録A.	90
付録B.	90

目次

2.1	重回帰モデルを用いた工数見積りの相対誤差分布例	6
2.2	工数種別で分割した場合の MRE 箱ひげ図	14
2.3	工数算入業務で分割した場合の MRE 箱ひげ図	15
2.4	FP 数を中央値で分割した場合の MRE 箱ひげ図	15
2.5	層別モデルと非層別モデル実験の比較手順（開発形態の場合）	19
2.6	開発形態での非層別，層別の MRE の比較	20
2.7	FP 数の非層別，層別の MRE の比較	21
3.1	提案するフィットデータ選定方法	29
3.2	適合度指標ごとの相関係数の箱ひげ図	39
3.3	RMS 順位 1 位モデル適用時の相対誤差順位のヒストグラムとその累積%	39
3.4	Adj.R ² 順位 1 位モデル適用時の相対誤差順位のヒストグラムとその累積%	40
3.5	提案方法とそれ以外の方法による見積り精度の箱ひげ図（MRE）	41
3.6	提案方法とそれ以外の方法による見積り精度の箱ひげ図（MER）	42
4.1	重回帰分析による見積り結果の一例	47
4.2	類似度の平均が小さい場合と大きい場合の例	49
4.3	類似プロジェクトの工数が小さい場合と大きい場合の例	50
4.4	ばらつきの指標ごとの $\text{Pred}(\overline{100})$ の変化（Desharnais データ，Euclidean Similarity）	57
4.5	ばらつきの指標ごとの $\text{Pred}(\overline{100})$ の変化（Kitchenham データ，Euclidean Similarity）	57

4.6	ばらつきの指標ごとの $\text{Pred}(\overline{100})$ の変化 (ISBSG データ , Euclidian Similarity)	58
4.7	ばらつきの指標ごとの $\text{Pred}(\overline{100})$ の変化 (Desharnais データ , Cosine Similarity)	58
4.8	ばらつきの指標ごとの $\text{Pred}(\overline{100})$ の変化 (Kitchenham データ , Cosine Similarity)	59
4.9	ばらつきの指標ごとの $\text{Pred}(\overline{100})$ の変化 (ISBSG データ , Cosine Similarity)	59
4.10	ばらつきの指標ごとの MdmRE の変化 (Desharnais データ , Euclidian Similarity)	60
4.11	ばらつきの指標ごとの MdmRE の変化 (Kitchenham データ , Euclidian Similarity)	60
4.12	ばらつきの指標ごとの MdmRE の変化 (ISBSG データ , Euclidian Similarity)	61
4.13	ばらつきの指標ごとの MdmRE の変化 (Desharnais データ , Cosine Similarity)	61
4.14	ばらつきの指標ごとの MdmRE の変化 (Kitchenham データ , Cosine Similarity)	62
4.15	ばらつきの指標ごとの MdmRE の変化 (ISBSG データ , Cosine Similarity)	62
4.16	見積り対象の工数実績・見積り値と類似プロジェクトの工数分散値との関係 (類似度 : Euclidian Similarity)	65
4.17	見積り対象の工数実績・見積り値と類似プロジェクトの工数分散値との関係 (類似度 : Cosine Similarity)	65
5.1	アナロジーベース法と複合法の $\text{Pred}(\overline{100})$ の比較 (Desharnais データ , Euclidian Similarity)	71
5.2	アナロジーベース法と複合法の $\text{Pred}(\overline{100})$ の比較 (Kitchenham データ , Euclidian Similarity)	72

5.3	アナロジーベース法と複合法の $\text{Pred}(\overline{100})$ の比較 (ISBSG データ, Euclidian Similarity)	72
5.4	アナロジーベース法と複合法の $\text{Pred}(\overline{100})$ の比較 (Desharnais データ, Cosine Similarity)	73
5.5	アナロジーベース法と複合法の $\text{Pred}(\overline{100})$ の比較 (Kitchenham データ, Cosine Similarity)	73
5.6	アナロジーベース法と複合法の $\text{Pred}(\overline{100})$ の比較 (ISBSG データ, Cosine Similarity)	74
5.7	アナロジーベース法と複合法の MdmRE の比較 (Desharnais データ, Euclidian Similarity)	74
5.8	アナロジーベース法と複合法の MdmRE の比較 (Kitchenham データ, Euclidian Similarity)	75
5.9	アナロジーベース法と複合法の MdmRE の比較 (ISBSG データ, Euclidian Similarity)	75
5.10	アナロジーベース法と複合法の MdmRE の比較 (Desharnais データ, Cosine Similarity)	76
5.11	アナロジーベース法と複合法の MdmRE の比較 (Kitchenham データ, Cosine Similarity)	76
5.12	アナロジーベース法と複合法の MdmRE の比較 (ISBSG データ, Cosine Similarity)	77

表 目 次

1.1	プロジェクトデータセットの一例	2
2.1	ISBSG データセットに含まれるプロジェクト特性	12
2.2	プロジェクト特性ごとの見積り精度	14
2.3	非層別モデルと層別モデルでの見積り精度の比較	20
3.1	実験用 ISBSG データセットに含まれるプロジェクトの特性	35
3.2	モデル選択指標ごとのモデル適合度と相対誤差の相関係数	38
3.3	提案方法とそれ以外の方法による見積り精度の比較 (MRE)	41
3.4	提案方法とそれ以外の方法による見積り精度の比較 (MER)	41
3.5	特性ごとのモデル選択割合	44
4.1	ばらつきの各指標と MRE との相関係数 (類似度: Euclidian Similarity)	56
4.2	ばらつきの各指標と MRE との相関係数 (類似度: Cosine Similarity)	56

第1章 はじめに

1. 研究の背景と概要

近年，開発されるソフトウェアの多品種化に伴い，開発プロセスが多様化している．例えば，エンタープライズ系と呼ばれる業務用ソフトウェアに限った場合においても，ソフトウェアが対象とする業務，開発に利用する業務パッケージの種類，アーキテクチャ，開発プラットフォーム，開発言語などが多岐にわたっており [45]，それらの違いによって開発プロセスは影響を受ける．このような現状のもとでは，開発プロジェクトの個別性を考慮し，個々の開発プロセスに合わせた開発計画の立案が必要である．そのベースとなるのが，開発に必要な労力（開発工数）の見積りであり，例えば，金融取引処理システムなど，高度な信頼性が要求されるシステムにおいては，信頼性を高めるための作業（設計レビュー，システムテストなど）により多くの工数を見積っておくことが求められる．ここでいう工数とは，開発要員数 × 時間で算出される，のべ作業時間をあらかず指標であり，単位は「人月」，「人日」，または「人時」である [8][11][18]．これまで工数見積りについての研究が数多く行われてきており，その1つに定量的工数見積り方法がある [2][4][7][33][34]．定量的工数見積り方法では，過去の開発プロジェクトで計測，収集されたデータを見積りの根拠として用いる．収集されるデータにはさまざまなプロジェクトの特性，例えば開発するソフトウェアの規模，開発に用いる言語などが記録されており，定量的工数見積りでは特性として記録された値を用いて見積りを行う．表 1.1 にプロジェクトデータセットの例を示す．

定量的工数見積りは見積りの根拠として過去プロジェクトの特性を利用するため，見積りの精度は過去のプロジェクトデータの集合，すなわちフィットデータセットの内容に大きく依存する．しかし，ソフトウェア開発はプロジェクトごと

表 1.1 プロジェクトデータセットの一例

ID	規模	開発言語	開発形態	...
0001	326	C	新規開発	...
0002	1050	Java	拡張開発	...
0003	130	Perl	新規開発	...
0004	950	C++	再開発	...
...

に特性が大きく異なる（個別性が高い）ため，あらゆる開発プロジェクトに適合する万能の見積り方法を定量的工数見積り方法から得ようとする，膨大な数の過去データが必要になり，現実には不可能である．

本論文では実用に耐える定量的工数見積り実現のために，従来広く用いられてきた工数見積り方法である線形重回帰モデルとアナロジーベース法に対し，開発プロジェクトの個別性を考慮した拡張をそれぞれ行うことで，平均的な見積り精度の向上と見積りを大きく外すプロジェクトの低減を実現する．

(1) プロジェクトの特性に応じたフィットデータ選定方法

重回帰モデルによる工数見積りでは，プロジェクトの特性を説明変数として用い，目的変数である開発工数との関係を表すモデル式を導出し，見積り対象のプロジェクトに用いる．このため，重回帰モデルによる見積りの精度はフィットデータセットに大きく依存する．しかしながら，従来の重回帰モデルによる見積り方法は見積り対象のプロジェクトの特性を考慮せず，全ての過去プロジェクトを用いて見積りを行っていた．そこで，重回帰モデルに基づく工数見積りに，プロジェクトの個別性を考慮してフィットデータを選定する手順を追加する．

このような選定としては，既に本間 [44] によって提示された方法が存在する．しかしながらその方法は選定が手作業によって行われるため，選定を行う際にはソフトウェア開発についての十分な知識と経験が必要となる．また，通常プロジェクトには複数の特性が含まれており，十分な知識がなければ，その中でどの特性

を用いれば最も精度が高くなるかは分からない。例えば、開発言語と開発種別（新規開発か改造開発か）のどちらを用いれば精度が向上するかを判別するのは難しい。さらに、あるプロジェクトに対して特性の中から最も精度が高くなることが期待できる1つが選べたとしても、その特性が他のプロジェクトの時にも使えるとは限らない。例えば開発言語が最も精度が高くなることが期待できることが分かったとしても、見積り対象のプロジェクトの開発言語と同じ言語で開発しているプロジェクトが過去のプロジェクトデータにほとんど含まれていない場合、少数のプロジェクトデータでモデルを構築することになる。そのようなモデルを用いて見積りを行っても、高い精度の見積りは期待できない。そこで、提案手法では見積り対象プロジェクトの特性ごとに過去プロジェクトから、同じ特性を含むプロジェクトをそれぞれ抽出し、フィットデータセット候補群とする。ただしこのとき、フィットデータに含まれるプロジェクトの件数が少ない場合、そのフィットデータは破棄する。これは、少数のプロジェクトから構築されたモデルは過学習が起りやすく、モデルによる見積りの結果を信頼できないためである [12]。破棄されずに残った候補群それぞれについてモデルを構築し、それらの中でモデルの適合度が最も高かったフィットデータを見積りに最適なフィットデータとして採用、見積る方法を提案する。

(2) 類似プロジェクトの特性に応じた見積り回避プロジェクトの選定方法

アナロジーベース法は、見積り対象の開発プロジェクトに類似するプロジェクト（通常は複数のプロジェクト群）を、過去の実績データの中から選定し、その実績工数を用いて開発工数を見積る方法である。重回帰モデルと比べて、開発プロジェクトの特性（個別性）を考慮した見積りが行える点に特長がある。しかしその一方で、類似プロジェクト群がうまく選定できない場合には、極端に見積り精度が悪くなることが課題であった。そこで、見積り対象のプロジェクトに「類似していると判定されたプロジェクト群が均質でなければ見積りに失敗しやすい」という仮定に基づき、類似プロジェクトの群のばらつきを表す指標（類似度の平均、分散や類似プロジェクトの工数の変動係数、分散）が大きい場合、見積り対象のプロジェクトをうまく見積りできない（見積りを大きく外す）と判断し、見

積り自体を回避する．そして回避されたプロジェクトについては重回帰モデルによる見積りを行うことで見積りを大きく外す可能性を低減する．このように，プロジェクトの個別性を考慮せず，どのようなプロジェクトに対してもアナロジーベース法を適用するのではなく，見積り対象プロジェクトの特徴に応じて見積り方法自体を変更する方法を提案する．

2. 論文構成

以降，2章では重回帰モデルによる工数見積りにおいて，どの特性を含むときにプロジェクトのばらつきが大きくなるかを調べ，見積りをばらつかせる原因となるプロジェクト特性を特定する．そして，ばらつきが大きいプロジェクト特性について，特性ごとにフィットデータを分割しモデルを構築することでばらつきを抑えることを目指す．続いて3章において，モデル構築に用いるフィットデータを，プロジェクト特性によって層別する方法を提案する．これは2章のばらつきを抑える方法を発展させたものである．フィットデータを見積り対象のプロジェクトが含む特性に合わせて複数に分割し，そのうちで最も精度が高くなると期待できるフィットデータを選択し，モデルを構築して見積りに用いるものである．これは1章の(1)に対応する．4章では，見積りのばらつきの原因となる，大きく見積りが外れる（見積りに失敗する）プロジェクトを特定するための方法を提案する．これはアナロジーベースな方法において，類似プロジェクトの工数や類似度のばらつきが大きい場合，見積りに失敗しやすいという仮定に基づくものである．続いて5章では，アナロジーベース法では見積りが失敗すると判定されたプロジェクトについて，別の方法，ここでは対数重回帰モデルを用いて見積りを行うことで，見積りが失敗する可能性を極力低減する方法を提案する．これによって4章では見積りに失敗するとして見積りから外されたプロジェクトについても，失敗する可能性が低い見積り結果を得ることが可能となる．これは1章の(2)に対応する．そして6章において本論文で得られた知見についてまとめる．

第2章 特性を考慮した工数見積りモデルの性能調査

1. まえがき

ソフトウェア開発プロジェクトの円滑な進行，完遂のために，プロジェクトの初期段階で工数見積りを行うことが重要である．工数見積りには工数見積りモデルを用いる方法が一般的であり，従来，重回帰モデル，ニューラルネット [34]，事例ベース推論 (CBR)[33] などの数多くの工数見積りモデルが提案されてきた．工数見積りモデルとは，プロジェクト特性（開発規模，開発者数など）を説明変数とし，目的変数である工数との関係を数学的に表すものである．過去のプロジェクトで計測された特性に基づいてモデルの構築を行い，現行のプロジェクトで計測した特性をモデルに代入することで工数の見積り値を算出できる．

しかし，重回帰モデルを用いた工数の見積り精度には，プロジェクトごとに大きなばらつきがある．1つの例として，ステップワイズ対数重回帰モデルにより68件のプロジェクトの試験工数の見積りを試みた結果を図2.1に示す [42]．図2.1は，工数見積り値の相対誤差の分布を示している．この例では，相対誤差が30%以内の「見積りがほぼ的中した」プロジェクトが数多く存在する一方で，100%を超える「見積りに失敗した」プロジェクトも少なからず存在する．このように，見積りが大きく外れる可能性がある以上，たとえ平均的には誤差が小さくとも，見積り値を信頼してプロジェクト管理を行うことは危険である．

このため，工数見積りモデルを現場で用いるためには，見積りが外れやすい（もしくは見積り精度がばらつきやすい）プロジェクトの特徴を明らかにしておくことが必要である．一つの典型的な例として，開発中に何らかのトラブルが発生して予定よりも工数が大きく超過したプロジェクトでは見積りは大きく外れること

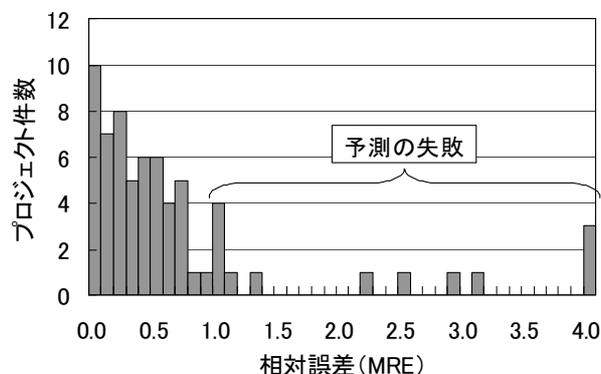


図 2.1 重回帰モデルを用いた工数見積りの相対誤差分布例

になる．トラブルを未然に防ぐために，様々なリスク評価方法が従来提案されてきた [13]．ただし，トラブルの発生以外にも，見積りが外れる要因は数多く存在する．例えば，外注率が高いために正確なデータの収集が難しい場合や，データ収集のための人員が不足している場合は，収集したデータ（プロジェクト特性）の信頼性が低くなり，正確な見積りが行えない可能性がある．また，開発方法や開発体制によっては見積り精度に大きなばらつきが出る可能性がある．例えば，Java 言語を用いた開発では（COBOL などと比較して）コードの自動生成を伴うことが多いため，ソースコード行数をソフトウェア規模として扱うと，見積りに用いると，精度が大きくばらつく可能性がある．

本章では，まず見積りが外れやすい，もしくはばらつきやすいプロジェクトの特徴を実験的に明らかにする．そして，明らかとなったばらつきやすい特徴をもつプロジェクトについて，そのばらつきを抑えるために見積りモデルの構築に用いる，フィットデータに対して層別を行うことを提案する．層別は事例選択の 1 種であり，事例選択は一般に見積りの精度を高めるといわれている．しかしながらこれまでソフトウェアの工数見積りについての研究で行われてきた事例選択は，例えば外れ値となるプロジェクトを除外する [47] といったものであり，特性による層別の有効性について検証した研究は筆者らの知る限り存在しない．実験では，まず重回帰分析により多数のプロジェクトの工数見積りを行い，各プロジェクトの見積り誤差を求める．ここでは設計終了時に総工数を見積る場合を想定した．

次に、プロジェクト特性ごとにプロジェクトを複数の群に分割する。プロジェクト特性が名義尺度の変数の場合には、その名前（カテゴリ；開発言語ならばCやJavaなど）ごとに分割し、比尺度の場合には中央値を閾値として2分割する。また順序尺度、間隔尺度は名義尺度として扱う。分割後にそれぞれのグループごとに見積り誤差に有意な差があるかの検定を行う。その結果、有意差が見られた組み合わせについてはその原因を考察する。さらに、有意差が見られたプロジェクト特性については、特性ごとにあらかじめフィットデータを分割し、分割したデータセットからモデルを構築、見積りを行う。その見積り誤差を先ほどの見積り後に分割を行った場合の結果と比較し、考察を行う。

以降、2章2節では本研究で用いた見積り方法であるステップワイズ重回帰分析について述べる。2章3節において、本研究で用いたデータセットであるISBSGデータセットの詳細について説明する。2章4節で工数見積り精度を低下させる要因を特定するために行った実験の概要、手順、結果を述べ、考察を行う。続いて2章5節では追加実験についてその概要、手順、結果を述べ、考察を行った。この追加実験は、特定のプロジェクト特性を含む実績データのみから構築した工数見積りモデルを用いて工数見積りを行う場合と比較する目的で行った。2章6節で関連研究について説明し、2章7節で本章のまとめを述べる。

2. ステップワイズ重回帰分析

本研究では、工数見積りの方法として、多変量解析の方法の一つであるステップワイズ重回帰分析を用いる。工数見積り方法としては、重回帰分析以外にもニューラルネット [34]、事例ベース推論 (CBR) [33] などの数多くの工数見積りモデルが提案されているが、重回帰分析はソフトウェア開発工数見積りモデルの作成において最も一般的に用いられているため、これを採用する [6] [37]。

重回帰分析では、目的変数と、それに対して影響すると考えられる複数の説明変数の関係を表す1次式を導出する。この1次式では、絶対誤差の2乗和が最小となる係数が与えられる。工数見積りにおいては、工数が目的変数であり、それ以外の特性が説明変数の候補となる。

ステップワイズ重回帰分析は，ステップワイズ変数選択法により説明変数を決定し重回帰分析を行う方法である．ステップワイズ変数選択法は，目的変数に対して強く影響する変数を説明変数として選択する方法の1つである．変数選択は以下の手順で行われる．

手順1 初期モデルを作成する．

手順2 作成されたモデルに対して，各説明変数の係数が0でないかの検定を行い，指定した有意水準で棄却されない場合に変数を採択し，棄却される場合には変数を除去する．本論文では多重共線性を回避するために，採択する変数の分散拡大要因が10以上の場合，またはその変数を採択することによって他の変数の分散拡大要因が10以上となる場合には，その変数は採択しないという操作を行う．

手順3 検定により適切な変数が選択されたと判断されるまで手順2を繰り返す．

本論文では変数選択法としては変数増減法を採用する．変数増減法では，変数を全く含まないモデルを初期モデルとして変数選択を行う．これ以外にも，ステップワイズ変数選択法としては変数増加法，変数減少法，変数減増法がある．しかしながら，変数増加法は一度追加された変数が削除されることがないため，ある段階では有用な変数であっても，その後，他の変数が取り込まれた段階では有用ではなくなる可能性があり，変数減少法でも同様に，いったん削除された変数が，他の変数が削除されているうちに有用になる可能性があるという問題がある．これらの問題が起きることのない変数増減法と変数減増法のうち，変数増減法を採用したのは工数見積りについての論文の多くで重回帰分析のための変数選択として用いられていること [5][23][33]，および，本論文で用いた代表的な統計パッケージの1つである SPSS でも変数増減法が採用されているためである．

3. ISBSG データセット

実験に用いたデータセットは International Software Benchmarking Standards Group (ISBSG) が収集した，20ヶ国のソフトウェア開発企業の実績データである

[11] . 工数の見積りに利用可能な実績データとしてはこれ以外にも , Kitchenham らが文献 [18] で公開しているもの , Desharnais らが文献 [8] で公開しているものなどが存在するが , ISBSG のデータセットはこれら 2 つのデータセットに比べて含まれるプロジェクトの数が多く , また収集されている特性の種類も豊富であることからこれを採用した . 後述する提案方法において , 特性による選定の効果を評価するためには十分な大きさのデータが必要であると考えられるためである . ISBSG が収集したデータセットは , 20ヶ国のソフトウェア開発企業の実績データによって構成されている . このデータセットには , 1989 年から 2004 年までの 3,026 件のプロジェクトについて , それぞれ 99 種類の特性値が記録されている . ただし , このデータセットには数多くの欠損値 (全プロジェクトにおけるプロジェクト特性の平均欠損率 : 57.6%) が含まれているため , 全てのデータをそのまま重回帰分析に用いることはできない .

本章では , ISBSG データセットから欠損を含まないサブセットを作成し , 実験用のデータセットとして用いた . これは重回帰分析は欠損値を含むデータを扱うことができないため , 欠損を含むデータセットに対してなんらかの処理を加える必要があるからである . 欠損値に対する一般的な対処法としては , 欠損値を別の値で補う方法 , 欠損を多く含むプロジェクト , プロジェクト特性を除外するという 2 つの方法が挙げられる . しかし欠損値を別の値で補う方法には , データセットの各特性値の平均や分散といった統計的特性が変化するという問題があり , さらに欠損値分布に偏りが生じている場合には , 見積り精度が著しく低下するなどの問題がある [19] . 欠損の有無が工数見積りの精度に影響を与えることは , 見積り精度のばらつきの原因となるプロジェクト特性を明らかにするという本章の目的を達成する上では望ましくない . このことから本章では欠損を多く含むプロジェクトおよびプロジェクト特性を除外したサブセットを作成し , 実験に用いた .

サブセットの作成では , まず ISBSG データセットから , 文献 [15] , 文献 [24] にならってデータクオリティレーティングが C または D のプロジェクトを除外した . データクオリティレーティングとは ISBSG に所属する専門家が , そのプロジェクトデータが信頼の信頼性を 4 段階で判定したものであり , 信頼できるならば A , 全く信頼できなければ D と記される . すなわち , ここではデータの信頼

性が低いと判定されたプロジェクトを除外した。次に FP 数の測定方法が IFPUG ではないものを除外した。これは実際には同規模のプロジェクトであっても測定方法によって FP 数が変化するおそれがあるためである。これらの処理を行った後に、例えば計画工数などの工数に関わる特性などの、致命的な致命的な欠損を含むプロジェクトや欠損しているプロジェクトが多い特性を除外した。このような手順を経て、欠損を含まない 114 件のプロジェクト、10 種類の特性を含むデータセットを作成し、実験に用いた。2 章 1 節で述べたとおり、本章では見積りのタイミングとしては設計終了時を想定しているため、この 10 種類の特性は全て設計終了時まで決定可能と考えられる変数のみを採用している。

これら 10 種類のプロジェクトの特性について、特性値名、尺度の種別、名義尺度に含まれるカテゴリ名の一部、そしてそれぞれの名義・順序尺度に含まれる特性値の種類数を表 2.1 に示す。10 種類の特性変数には 4 種類の比尺度変数、5 種類の名義尺度変数、1 種類の順序尺度変数が混在している。このうち名義尺度変数および順序尺度変数については、それぞれ 0 または 1 の値をとる複数の 2 値変数（ダミー変数）に変換したものを用いた。ダミー変数への展開手順を付録 A に示す。表 2.1 中の「その他」とは、名義尺度において含まれるカテゴリが 4 種類以上の場合に、あるカテゴリを含むプロジェクトの数が 10 以下のものを 1 つにまとめたものである。これは、あまりにも件数が少ないカテゴリは有意差検定に用いても信頼できる結果が得られないと判断したためである。

ここで工数計測方法に含まれるカテゴリである実作業時間集計と報告作業時間集計、工数算入業務に含まれるカテゴリであるレベル 1~4 について詳しく説明する。工数計測方法と工数算入業務は、どちらも工数の測定方法に関係する名義尺度である。工数は人員 × 作業時間の値であり、その算出には人的要素と時間的要素が必要である。このうち人的要素に関わるのが工数算入業務であり、時間的要素に関わるのが工数計測方法である。

工数算入業務とは、プロジェクトとしての作業内容の定義であり、どのような業務に従事する「作業員」をプロジェクトの開発に参加していると認定し、工数として報告させたかを表している。これに対して工数計測方法は作業時間の記録方法の定義であり、どのような業務に費やした「時間」をプロジェクトの開発と

して認定し，報告したかを表している．

工数算入業務のレベルごとの具体例を以下に示す．まずレベル1とは，開発チーム（プロジェクトチーム，プロジェクトマネジメント，プロジェクト管理）が業務に費やした時間を作業時間として認めるものである．以下同様に，レベル2は開発チームを直接サポートした人員（データベース管理，データ管理，品質保証，データセキュリティ，標準化サポート等）が，レベル3ではコンピュータオペレーションなどの開発チームを間接的にサポートした人員（ソフトウェアのサポート，ハードウェアのサポート，ネットワーク管理等）が，レベル4ではエンドユーザやクライアントをサポートした人員（ユーザやクライアントへの教育，情報伝達等）が業務に費やした時間を作業時間として認め，工数に算入している．ただし，高いレベルの工数算入業務はそれより低いレベルの工数算入業務を含んでいる．例えばレベル2はレベル1の定義内容を含んでおり，レベル3はレベル1とレベル2の定義内容を含んでいる．レベル4も同様である．次に工数計測方法について述べる．まず実作業時間集計とは，純粹にプロジェクト開発に費やした時間のみを作業時間として記録し，工数の算出に用いるものである．この場合，教育などのプロジェクト開発に直接には関わっていない作業に費やされた時間は一切工数に含まない．これに対して報告作業時間集計とは，プロジェクトに関する作業であれば，実作業時間集計では除外される業務，例えば教育等の作業であっても，それに費やされた時間を工数として記録し，工数の算出に用いるものである．工数計測方法と工数算入業務を含む ISBSG データセットの全ての特性についての，より詳細な説明については文献 [10] を参照されたい．

4. 工数見積りモデルの精度を低下させる要因の特定

4.1 特定のための実データを用いた実験

4.1.1 概要

工数見積りの見積りが外れやすい，または見積り精度がばらつきやすいプロジェクトの特徴を明らかにするための実験を行った．まず重回帰モデルを用いて

表 2.1 ISBSG データセットに含まれるプロジェクト特性

プロジェクト特性	含まれる特性
FP 数	数量データ
総工数	数量データ
計画工数	数量データ
要件定義工数	数量データ
開発形態	新規開発, 拡張開発, 再開発
アーキテクチャ	クライアントサーバ型, スタンドアロン型, 複合型
開発対象	メインフレーム, ミドルウェア, PC
開発言語	C, COBOL, PL/I, その他
工数計測方法	実作業時間集計, 記録作業時間集計, その他
工数算入業務	レベル 1~4

工数の見積りを行い, プロジェクト特性の違いが実績値と見積り値の差にどのような影響を与えるのかを調べた. 実験のフィットデータとテストデータの分割にはリーブワンアウト法を用いた.

4.1.2 手順

実験の手順は次の通りである.

手順 1 114 件のプロジェクトから 1 件を選んで見積り値算出用のテストデータとし, 残りの 113 件をモデル構築用のフィットデータとする.

手順 2 フィットデータを用いて重回帰モデルを構築する.

手順 3 テストデータに構築したモデルを適用し, 書くプロジェクトについて総工数の見積り値を算出する.

手順 4 算出された総工数見積り値と総工数実績値の差から相対誤差 (MRE: Magnitude of Relative Error) を算出する. MRE は実績値を E , 見積り値を \hat{E}

とするとき，式 2.1 で定義される [7] .

$$MRE = \frac{|E - \hat{E}|}{E} \quad (2.1)$$

手順 5 手順 2~4 を全てのプロジェクトが見積り対象として採用されるまで行う .
本実験の場合は 114 回となる .

手順 6 手順 1~5 の結果で得られた相対誤差とプロジェクトのセットについて ,
プロジェクト特性ごとにプロジェクトを 2 つ以上の群に分割し , 群間で相対
誤差についての有意差検定を行う .

分割は , 比尺度変数は中央値を閾値として大小の 2 群に , 名義尺度変数と順序尺度
変数についてはカテゴリごとに行った . 検定方法は 2 群の場合には Mann-Whitney
の U 検定を , 3 群以上の場合には Kruskal-Wallis 検定を用いた .

4.1.3 結果

実験終了後の検定の結果 , 名義・順序尺度変数では開発形態と工数算入業務に ,
比尺度変数では全てについて $p < 0.05$ で有意差が確認できた .

これらの結果について , その MRE の中央値 (MdMRE) , 平均値 (MMRE) ,
分散 (VMRE) を表 2.2 に , 名義尺度変数の全てと比尺度変数のうち FP 数の結
果の箱ひげ図を図 2.2 ~ 図 2.4 に示す . 箱ひげ図では縦軸が相対誤差 MRE を示し
ており , 図中の \square と * は外れ値を示している . 箱ひげ図において比尺度変数の結
果を FP 数のみとしたのは , それ以外の比尺度変数でもおおむね似た結果が得ら
れたためである . 箱ひげ図中ではサンプル数が少なかった開発形態が再開発の場
合 (2 件) と工数算入業務がレベル 3 の場合 (1 件) については省略した . また可
読性の都合上 , 縦軸 (相対誤差) の最大値を 3 (300%) までとしたが , それより
も大きな相対誤差も存在している .

表及び図から , まず名義尺度の特性のうち , 開発形態では新規開発の方が拡張
開発よりも精度良く見積りができており , 工数算入業務でもレベル 1 , 4 の方が
レベル 2 よりも明らかに高い精度で見積りができていることが分かる . 比尺度の
特性では , 全て上位の方が下位よりも高い精度で見積れていることが分かる .

表 2.2 プロジェクト特性ごとの見積り精度

プロジェクト特性	カテゴリ名	MdMRE	MMRE	VMRE
開発形態	新規開発	0.386	0.632	0.919
	拡張開発	0.643	1.313	4.120
工数算入業務	レベル1	0.479	0.803	2.053
	レベル2	0.882	1.887	5.090
	レベル4	0.422	0.573	0.634
FP 数	上位	0.356	0.435	0.162
	下位	0.710	1.438	4.305
計画工数	上位	0.669	0.455	0.181
	下位	0.669	1.419	4.324
要件定義工数	上位	0.415	0.400	0.079
	下位	0.738	1.473	4.315

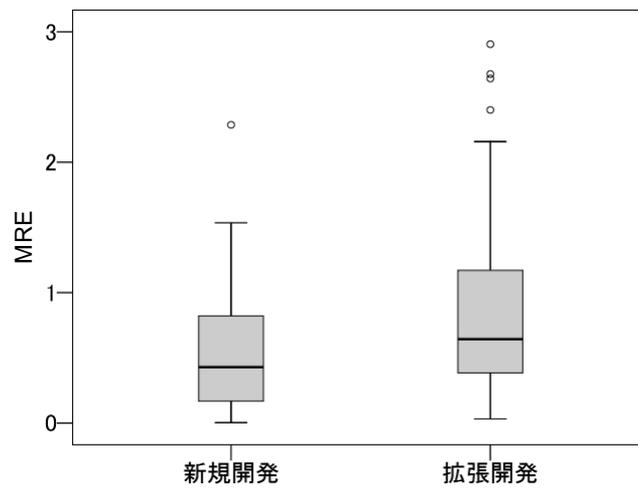


図 2.2 工数種別で分割した場合の MRE 箱ひげ図

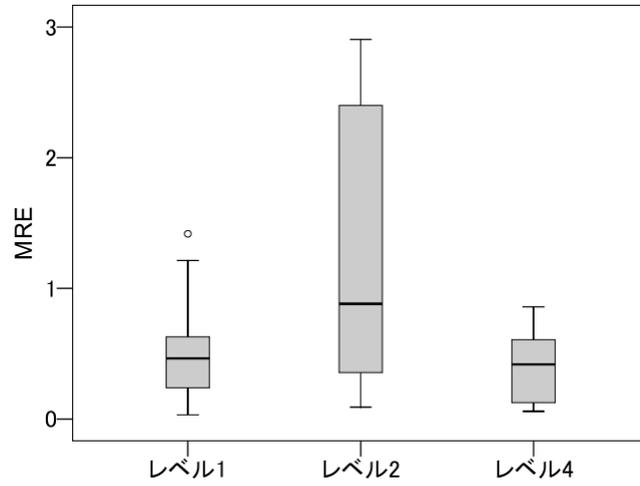


図 2.3 工数算入業務で分割した場合の MRE 箱ひげ図

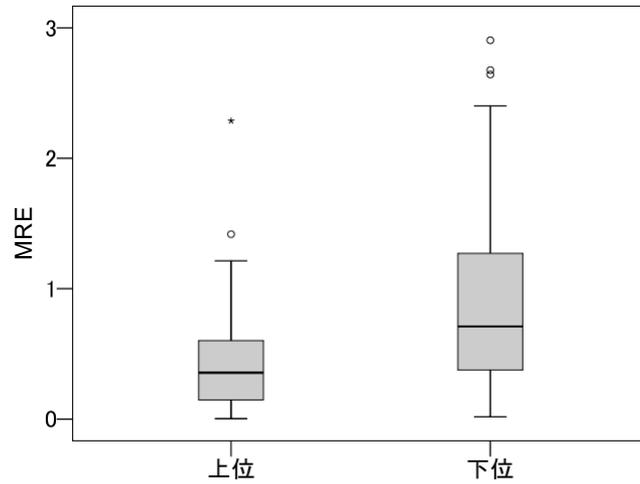


図 2.4 FP 数を中央値で分割した場合の MRE 箱ひげ図

4.1.4 考察

実験の結果，5種類の群について有意差が見られた．以下ではその理由を考察する．

まず開発形態で有意差が見られた原因としては，拡張開発は新規開発とは違い，開発済みのソフトウェアを対象に作業を行う．このため元となるソフトウェアの質が低ければ，コーディング等の作業に入る前に行う作業（前作業）に大きな工数を必要とする場合がある．このために拡張開発は新規開発に比べて規模に対する総工数のばらつきが大きくなり，この差が有意差として表れたと考えられる．

次に工数算入業務で有意差が見られた原因としては，レベル1ではほぼ開発チームのみ，レベル4では少しでもプロジェクトに関わった全ての開発者を「開発に携わった」として工数算入対象にする，という非常に明確な定義づけがされている．これに対して，レベル2やレベル3では「開発チームのサポート」や「ソフトウェア，ハードウェアのサポート」というように定義が曖昧であり，この結果，報告された業務レベルで定義されている業務内容と実際に行っている業務内容が異なっているという状況が起き，このためにばらつきが大きくなったと考えられる．FP数を中央値で2分割したときに有意差が見られた原因としては，重回帰分析は残差の2乗和を最小にするように係数を決定する．このため，目的変数の値が大きいプロジェクトの誤差がより小さくなるように最終的な回帰式が決定される．そして，FP数すなわちソフトウェアの規模は総工数と深い関わりがあるため，FP数の中央値で2分割した場合，回帰式は値の大きい方では誤差が小さく，小さい方では誤差が大きくなったものと考えられる．計画工数，要件定義工数についても同様である．

これらの結果から，開発形態では拡張開発が，工数算入業務ではレベル2が，FP数，計画工数，要件定義工数ではその値が中央値よりも小さいものが精度のばらつきと低下の要因となる特性であると考えられる．この結果を踏まえ，精度が低かった群については，その精度を高める方法を考える．例えば工数算入業務では，全てのプロジェクトを用いてモデルを構築した場合，レベル1やレベル4では高い精度で見積りができたが，レベル2では精度のばらつきが非常に大きく，結果を信頼できない事が分かった．この場合，レベル2については全てのプロジェ

クトを用いるのではなく，レベル2のプロジェクトだけを用いて重回帰モデルを構築し，見積りを行った方が精度が向上する可能性がある．また，既に高い精度で見積りができているレベル1，4についても，さらに精度を向上させることができる可能性がある．そこで，次節では，あらかじめプロジェクトの特性ごとにデータセットを分割し，モデルを構築，見積りを行った結果と，本章での実験結果を比較し，重回帰モデルを用いた場合の見積り精度の向上方法について検討する．

5. プロジェクト特性による層別モデルと非層別モデルの見積り性能の比較

5.1 性能比較実験

5.1.1 概要

2章4節で行った過去プロジェクト全てを用いて，工数見積りをばらつかせるプロジェクトの特徴を分析する実験（以下，非層別モデル実験と呼ぶ）の結果と考察から，全ての過去プロジェクトデータを用いてモデルを構築した場合，プロジェクトの特性によって見積り精度に大きな差が現れることが分かった．そして，全てのプロジェクトを用いてモデルを構築するよりも，特性ごとにデータセットからプロジェクトを抽出し，モデルを作成したほうが工数見積りの精度は向上する可能性がある．

本節では，プロジェクト特性ごとのフィットデータ分割による見積り精度向上の可能性を検証する．具体的には，全てのプロジェクトを用いてモデルを構築，見積りを行い，その結果をプロジェクト特性ごとに分割したものと，プロジェクト特性ごとにデータセットに対して層別を行い，層別後のデータセットに対してモデルを構築，見積りを行った結果とを比較する．ただし全てのプロジェクト特性について比較を行うと結果が煩雑になるため，ここでは非層別モデル実験で有意差が見られたプロジェクト特性についてのみ扱うことにする．

5.1.2 手順

重回帰モデルの構築における変数選択法等の実験環境，条件は非層別モデル実験と同一である．また簡便のため，本実験を層別モデル実験と呼称する．層別に用いるプロジェクト特性には，非層別モデル実験で有意差が見られた工数種別，工数算入業務，FP 数，計画工数，要件定義工数を採用する．実験手順は以下のとおりである．

手順 1 非層別モデル実験で用いたデータセットに対してプロジェクト特性で層別を行う．

手順 2 層別後のデータセットそれぞれについてリーブワンアウトで工数見積りを行い，相対誤差 MRE を算出する．

手順 3 非層別モデル実験で得られた全てのプロジェクトの見積り結果を，プロジェクト特性ごとに層別する．

手順 4 手順 2 で得られた結果と手順 3 で得られた結果を比較する．

手順 5 手順 1～4 を，層別に用いる全てのプロジェクト特性について行う

層別では，例えば開発形態では新規開発のプロジェクトだけを抽出して1つのデータセットとし，拡張開発についても同様に抽出し1つのデータセットとした．また非層別モデル実験と同様に，件数が極端に少ないプロジェクト特性については無視した．工数種別の場合を例に取った実験手順を図 2.5 に示す．図中では再開発を省略している．

5.1.3 結果

非層別モデル実験で得られた結果と層別モデル実験で得られた結果をプロジェクト特性ごとに表 2.3 に示す．また，代表的な例として，名義尺度のプロジェクト特性からは開発形態を，比尺度のプロジェクト特性からは FP 数の比較結果を箱ひげ図として図 2.6，2.7 に示す．

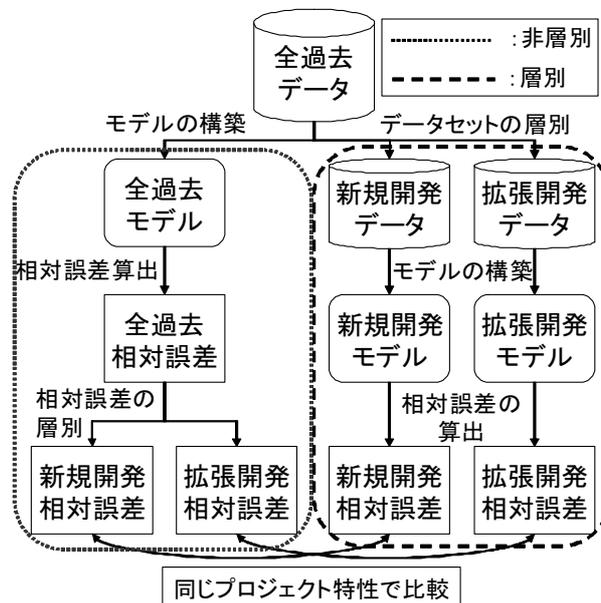


図 2.5 層別モデルと非層別モデル実験の比較手順（開発形態の場合）

それぞれのプロジェクト特性で非層別モデルの MRE と層別モデルの MRE について Wilcoxon 検定を行った結果，工数算入業務レベル 2 以外の全ての特性で有意水準 5% で差が確認できた．また有意差が確認できなかった工数算入業務レベル 2 では $p = 0.084$ であった．

表 2.3 から，MdMRE，MMRE の結果全てに共通する傾向として，非層別モデル実験で精度が高かったものは層別モデル実験では低く，非層別モデル実験で精度が低かったものは層別モデル実験では高くなるという，それぞれ逆の結果となっていることが分かり，このことは図からも分かる．VMRE でも，非層別モデル実験で精度が高かった特性については同様の傾向が見られたが，そうでないものについては低下しているものがいくつか見られた．

5.1.4 考察

表 2.3 および図 2.6，2.7 の結果から，MdMRE，MMRE については層別モデルは非層別モデルで高い精度を示した特性では精度が低下するが，非層別モデルで低い精度を示した特性では精度が向上することが分かった．VMRE では一部の

表 2.3 非層別モデルと層別モデルでの見積り精度の比較

		MdMRE		MMRE		VMRE	
		非層別	層別	非層別	層別	非層別	層別
開発形態	新規開発	0.386	0.409	0.632	0.786	0.919	1.623
	拡張開発	0.643	0.588	1.313	0.820	4.120	0.827
工数算入業務	レベル1	0.479	0.528	0.803	1.029	2.053	4.090
	レベル2	0.882	0.472	1.887	1.330	5.090	9.256
	レベル4	0.422	0.602	0.573	1.391	0.634	4.934
FP 数	上位	0.356	0.422	0.435	0.759	0.162	0.871
	下位	0.710	0.221	1.438	0.547	4.305	0.988
計画工数	上位	0.393	0.473	0.455	0.589	0.181	0.338
	下位	0.669	0.411	1.419	0.970	4.324	4.424
要件定義工数	上位	0.415	0.490	0.400	0.511	0.079	0.121
	下位	0.738	0.402	1.473	0.675	4.315	1.537

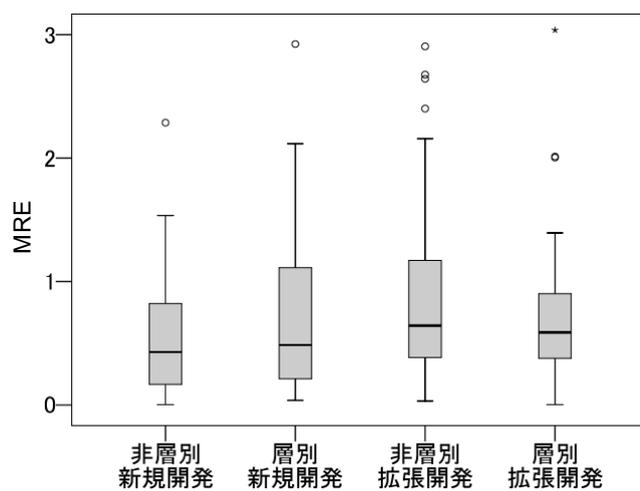


図 2.6 開発形態での非層別，層別の MRE の比較

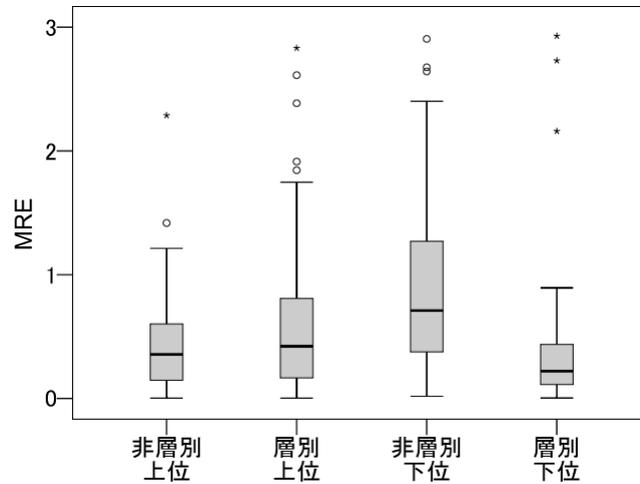


図 2.7 FP 数の非層別，層別の MRE の比較

特性について非層別モデルで精度が低かったものがさらに低下していたが，それらの特性でも MdMRE および MMRE の精度は向上していたことから，その原因は，一部の大きく見積りが外れたプロジェクトによって分散の値が大きくなったためであると考えられる。

比較結果から，プロジェクトの特性には全ての過去プロジェクトを用いてモデルを構築したほうが精度が高くなるものと，プロジェクト特性ごとに過去のプロジェクトを選定し，モデルを構築したほうが精度が高くなるものの2種類があることが分かった。このことから，非層別モデルのように全てのプロジェクトに対して同一のモデルを適用したり，全てのプロジェクト特性ごとにそれぞれモデルを構築し，適用するよりも，見積り対象のプロジェクトが持つ特性ごとに適切なモデルを選択できれば見積り精度が向上する可能性があると考えられる。しかしながら，適切なモデルを選択することを考えるならば，「開発形態が新規開発」かつ「工数算入業務レベル2」といったプロジェクトでは，全ての過去プロジェクトを用いてモデルを構築した方がいいのか，それとも工数算入業務レベル2のプロジェクトだけを集めてモデルを構築した方がいいのかについて明確に判断を下すことは難しく，検討を要すると考えられる。

6. 関連研究

これまでに見積りが外れる要因を分析した研究がいくつか行われている [31] . これらの多くはインタビュー結果に基づいた分析であるのに対し, 本研究はプロジェクト特性に基づく分析を行っている点が異なる. Mizuno ら [26] は, コストの実績値が見積り値よりも超過したプロジェクトについて分析を行っている. Mizuno らの研究では, $| \text{実績コスト} - \text{見積りコスト} | \div \text{見積りコスト}$ を誤差と定義している. そして, 完了したプロジェクトについて, プロジェクトの計画 (WBS, PERT 図など) や執行 (ソフトウェアレビューなど) が, どの程度厳格に行われたかを, プロジェクトマネージャに得点付けしてもらい, その得点の合計と, 誤差との相関を調べている.

Jørgensen ら [31] は, 見積り工数に誤差が発生した理由に対して, 影響を及ぼしている要因について分析を行っている. あるソフトウェア開発企業において, それぞれ立場が異なる責任者 (プロジェクトマネージャや UI 開発部門の責任者等) にインタビューを行い, 68 個のプロジェクトの特徴と誤差との関係について分析を行っている. その結果は, 誤差に影響を及ぼす要因は回答者の立場, データの収集方法, データの分析方法にあるとしている. この研究は, インタビューの結果に基づいた分析を行う際の問題点について分析したものである.

プロジェクト特性に基づき, 見積りが外れる要因を分析した研究も少数ながら存在する. Gray ら [9] は, モジュールのタイプ, および特徴から, 工数見積りが過少見積り, 適正見積り, 過大見積りのいずれになりやすいかを分析している. ただし, この研究は, 熟練者による見積りと実績値との差を分析している点が本研究と異なる.

7. まとめ

本章では, まず見積りが外れやすい, またはばらつきやすいプロジェクトの特徴を明らかにするための実験を行った. 全ての過去プロジェクトを用いて工数の見積りを行い, その結果をもとにプロジェクト特性ごとにプロジェクトを2つ以上

のグループに分割した。プロジェクト特性内のカテゴリ間での見積り誤差について有意差検定を行った結果、いくつかのプロジェクト特性では有意差がみられた。

次に過去プロジェクトをあらかじめプロジェクト特性ごとに分割し、工数見積りモデルを構築、見積りを行った。このとき、分割に用いたプロジェクト特性は先の実験で有意差が見られたものに限定した。得られた結果と先の実験の結果を比較した結果、先の実験で精度が高かったものでは精度は低下し、精度が低かったものでは向上した。このことから、プロジェクト特性ごとに適切なモデルを選択することで見積り精度を高めることが可能であると考えられる。今後、見積り対象のプロジェクトに含まれるプロジェクト特性をもとにデータセットを選定し、適切な工数見積りモデルを構築するための方法について研究を行う。

第3章 ソフトウェア開発工数見積りのためのフィットデータ選定方法

1. まえがき

ソフトウェア開発プロジェクトの計画立案，開発要員の割り当てなどにおいて，定量的工数見積り方法が広く利用されている [4] [6]．定量的工数見積り方法では，過去のプロジェクトの実績データを見積りの根拠として用いる [7]．定量的工数見積り方法はモデルベース推論とメモリベース推論に大別できる．重回帰分析 [48] をはじめとするモデルベース推論では，プロジェクトの特性変数（開発規模，欠陥数など）を説明変数として用い，目的変数である開発工数との関係を表すモデル式を導出し，見積り対象のプロジェクトに用いる．これに対し，アナロジーベース法 [4] [36] をはじめとするメモリベース推論では，見積り対象プロジェクトに類似した過去のプロジェクト群を選定し，それらの実績工数を用いて（加重平均などにより）見積り値を得る方法である．見積りモデルの性能は，モデル式の導出に用いる過去のプロジェクトデータセット，すなわちフィットデータセットの内容に大きく依存する．

本章では，モデルベース推論の代表的な方法である重回帰モデルにおける問題を取り扱う．性能の良いモデルを得るためには，見積り対象のプロジェクトと開発環境や開発プロセスの類似するプロジェクトのみを選定し，フィットデータとして用いることが重要となる．例えば，フィットデータにメインフレーム系の開発プロジェクトと，組み込みソフトウェアの開発プロジェクトを混在させるべきではない．メインフレーム系の見積りを行いたい場合は，フィットデータにはメ

インフレーム系のみを含めることが望ましい。

ところが、ソフトウェア開発は本質的に個別性が高いため、開発プロセスや業種、アーキテクチャ、開発言語などの全てが一致するプロジェクトのみをフィットデータとして選定すると、プロジェクト件数が極端に少なくなり、モデルの見積りの性能がかえって低下する場合がある。このような場合には性質の異なるプロジェクトもフィットデータに含めざるを得ない。ところがどこまでをフィットデータに含めるべきであるかについての一般的な基準はなく、従来ほとんど研究されていない。そのためフィットデータの選定はモデル構築者の経験に任されてきたのが現状である。

本章では、重回帰モデルの構築において、見積り対象のプロジェクトの特性に応じてフィットデータを自動的に選定する方法を提案する。提案方法では、複数の特性を持つ見積り対象プロジェクトに対し、特性が1つ以上一致する過去プロジェクト群を選び出し、それぞれをフィットデータの候補とする。例えば、見積り対象のプロジェクトが業種 = 金融、言語 = C、アーキテクチャ = 2層クライアントサーバであるならば、業種 = 金融の過去プロジェクトだけを集めたフィットデータセット候補を作成し、同様に、言語 = Cのもの、アーキテクチャ = 2層クライアントサーバのものを作成する。さらに業種 = 金融かつ言語 = Cの過去プロジェクトだけを集めたものを作成し、同様に言語 = Cかつアーキテクチャ = 2層クライアントサーバのもの、業種 = 金融かつアーキテクチャ = 2層クライアントサーバのもの、そして3つの特性全てが一致するものを作成する。ただし、プロジェクト件数が非常に少なく、モデルを構築しても信頼性が低いと考えられるフィットデータ候補はこの時点で破棄する。次に、破棄されなかったフィットデータ候補群を用いて、それぞれ見積りモデルを構築する。そして、見積り対象のプロジェクトに対して最良の結果を与えると思われるフィットデータ候補およびモデルを、モデル構築時の適合度（フィットデータに対するあてはまりの良さ）に基づいて決定する。ただし、見積り対象プロジェクトに含まれる特性全てが過去のプロジェクトにほとんど含まれていない場合、構築されたモデル群の適合度によってはプロジェクト特性を考慮しないモデル（全ての過去プロジェクトのデータを用いたモデル）の方が適合度の面で優れていることがありえる。その場合に

は，全プロジェクトをフィットデータとして採用する．

提案方法の有用性を評価するために，一般に公開されている最も大規模なプロジェクトデータセットである International Software Benchmarking Standards Group (ISBSG) が収集したプロジェクトデータ (ISBSG データ [11]) を用いて実験を行った．ISBSG データは，1989年から2004年の間に世界20ヶ国のソフトウェア開発企業から収集され，見積り方法の有効性を評価する研究で広く用いられている [15][24][32]．

以降，3章2節ではモデル構築におけるフィットデータ選定の問題を整理する．続いて3章3節で提案方法の詳細について説明し，3章4節では提案方法の有効性を評価するための実験とその結果，そこから導かれる考察を述べる．3章5節で本章のまとめと今後の課題を述べる．

2. 重回帰モデル構築におけるフィットデータ選定の問題

多くのソフトウェア開発企業では，多種多様なソフトウェアの開発を行っており，開発プロジェクトごとに工数の見積りを行うことが必要である．その1つの方法は，過去のプロジェクトで収集されたデータを用いて重回帰分析等により工数見積りモデルを構築し，見積りを行うことである．

ただし，多種多様なソフトウェアの開発工数を1つのモデル式で表現するには限界がある．例えば，航空電子工学関連システムと社内業務システムを比較すると，同じ1万 SLOC (Source Lines Of Code) 規模のプロジェクトであっても開発のプロセスが大きく異なり，生産性に8~18倍の違いが生じる [22][29]．この違いは，開発工数の差となって表れるが，この差を一つの重回帰モデルが吸収することは難しい．重回帰モデルでは，システム種別の違いは0または1をとる2値の説明変数として与えることができるが，モデル式の性質上，2値の違いが工数に対して8~18倍の影響を与えるというような，乗法的な影響を表現することはできない．重回帰モデルでは，各説明変数は，目的変数に対して加法的に作用するためである [48]．

また一般に，過去の開発プロジェクトデータは，工数に影響を与える全ての変

数を含んでいるとは限らず、このことも多種多様なプロジェクトを一つのモデル式で表現することを困難にしている。例えば「システムに要求される信頼性」という変数は、テスト工数に大きな影響を与えるが、変数として計測されていないこともある。システムに要求される信頼性は、システム種別と関わりが深いいため、システム種別ごとに個別に工数見積りモデルを構築した方が、全プロジェクトを用いたモデルよりも高い精度が期待される。

以上のことから、性能のよい重回帰モデルを構築するためには、見積り対象のプロジェクトと特性が類似する、例えばシステム種別や開発環境などが類似しているプロジェクトのみをフィットデータとして選定し、モデル構築に用いることが重要となる。ただし、ソフトウェア開発は本質的に個別性が高いため、システム種別、業種、アーキテクチャ、開発言語などの全ての特性が一致する類似プロジェクトのみをフィットデータとして選定すると、プロジェクト件数が極端に少なくなり、モデルの見積り性能がかえって低下する事例が報告されている [40]。このような場合には、性質の異なるプロジェクトもフィットデータに含めざるを得ない。

ところが、どの特性に着目してフィットデータを選定すべきかの一般的な基準はなく、従来ほとんど研究されていない。そのためフィットデータの選定はモデル構築者の経験に任されてきたのが現状である。たとえば本間 [44] は過去に行われた 1,050 件のプロジェクトのデータの分布状況を確認し、「新規開発・一括請負・Web 系開発」や「新規開発・一括請負・ビジュアル開発ツール」といったプロジェクト・タイプ約 30 種類に絞り込みを行ったうえで回帰分析を行い、数学モデルを抽出した事例を紹介している。ただしこの事例では、プロジェクト・タイプの分類を決定するにあたっては、データの分布を目視してモデルの精度を上げていくと述べられており、専門家やその作業に専従する開発経験の豊富な分析者の関与が必要となる。また本間の事例では、全てのプロジェクトはいずれかのタイプに分類され、見積りが行われることを暗に想定しているが、プロジェクトは多数の特性を持つことから、プロジェクト・タイプは常に互いに独立しているとは限らない。例えば「新規開発・一括請負・Web 系開発」と「新規開発・一括請負・規模 = 大」は重複がある。このような分類の取り扱いについては、従来決まった方

法が提案されていない。

3. 提案するフィットデータ選定方法

本章では、見積り対象プロジェクトと一致する特性を1つ以上持ち、かつ見積り精度が高くなることが期待できるフィットデータを過去のプロジェクトから自動的に選定する方法を提案する。多くの開発現場では、どのようにして最適な見積りモデルを得るかが課題となっている。提案方法は最適見積りモデルは最適なフィットデータから構築される、という立場からこの問題に1つの解決策を与える。本章で提案する方法は、あらかじめ（プロジェクトの重複を許さない）独立したプロジェクト・タイプ群を決めるのではなく、見積り対象プロジェクトごとにその特性に応じてフィットデータを選定する点が本間 [44] の方法と異なる。また提案方法は、常にフィットデータを選定するのではなく、プロジェクトによっては、全データを用いることも可能としている。

図 3.1 を例に提案方法の手順を説明する。図 3.1 では、見積り対象プロジェクトは言語 = COBOL，業種 = 金融，開発形態 = 再開発である。手順は下記の通りである。提案方法では一致する特性の数を3つ以上とすることが可能であるが、説明の都合上、ここでは一致する特性の上限数が2つの場合を述べる。3つ以上の場合に適用可能な、より一般化された手順の詳細を付録 B に示す。

- [手順1] 過去の全プロジェクトを集めたデータセット（以下、全過去データ）から、見積り対象プロジェクトの特性ごとに値が一致するプロジェクト群を選定し、それぞれをフィットデータ候補とする。例えば、開発言語という特性について、値がCOBOLであったプロジェクトを選定し「言語 = COBOL データ」とする。同様の作業を業種が金融（業種 = 金融データ）のもの、開発形態が再開発（開発形態 = 再開発データ）であるものについても行う。さらに複数の特性の組み合わせについてもこの作業を行う。具体的には、まず開発言語がCOBOLかつ業種が金融であるプロジェクトを選定し「言語 = COBOL かつ業種 = 金融データ」とする（図中COBOL金融データ）。他の組み合わせについても同様の作業を行い、「業種 = 金融かつ開発形態 = 再開

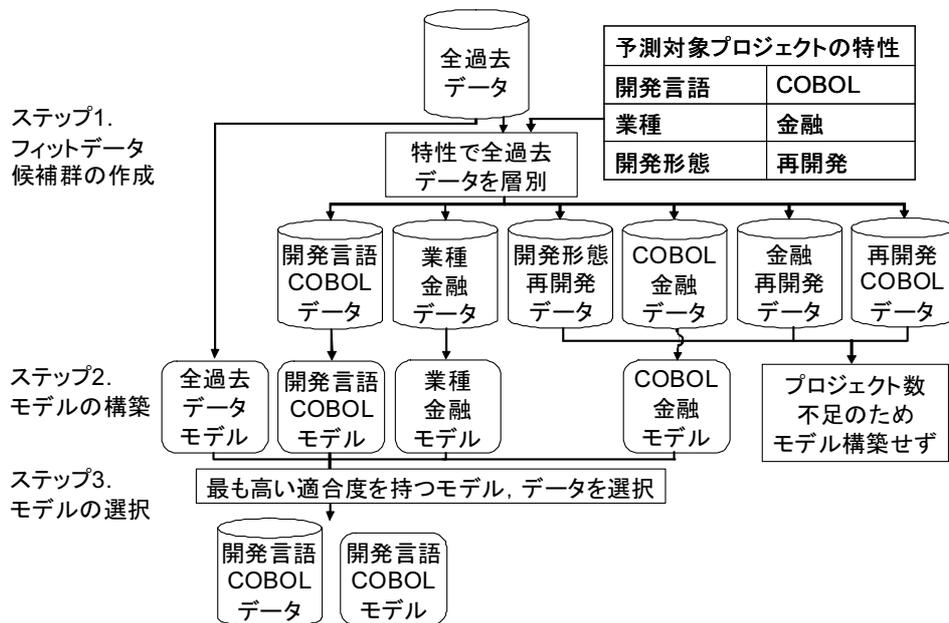


図 3.1 提案するフィットデータ選定方法

「開発言語 = COBOL データ」および「開発形態 = 再開発かつ言語 = COBOL データ」を得る。ただし、選定されたデータセットに含まれるプロジェクトの件数が一定数（以降、最小フィットデータサイズと呼ぶ）に達しておらず、モデルを構築するのに十分でない、または構築しても信頼できないと考えられる場合にはそのデータセットは破棄する。図中「開発形態 = 再開発データ」およびこれを含む組み合わせがこれに該当する。破棄されなかったデータセット群に全過去データを加えたものを、最終的なフィットデータ候補群とする。全過去データを候補に加える理由は、選定後のデータが全て破棄される場合があることと、特性ごとに選定されたフィットデータから構築したモデルよりも全データを用いたモデルの方が高い精度が得られる場合があるためである。

[手順2] フィットデータ候補群それぞれについて重回帰モデルを構築し、モデルのフィットデータに対する適合度を表す指標（残差平方平均，調整済み決定係数など）を記録しておく。

[手順3] 適合度を表す指標を基に，最も高い精度での見積りが期待されるモデルおよびそのフィットデータを選定する．図 3.1 の例では，開発言語 = COBOL データおよびモデルが選択されている．このデータおよびモデルを，最適なフィットデータおよびモデルと見なす．

[手順4] ステップ3で最適と見なされたモデルを見積り対象プロジェクトに適用し，見積り値を得る．

以上のステップは説明の簡略化のために，見積り対象プロジェクトの全ての特性が名義尺度変数として与えられる場合の手順を述べたものである．現実にはプロジェクト特性には開発期間，ファンクションポイント (FP) などの比尺度変数も含まれるため，より複雑な手順となる．具体的には，比尺度変数については，閾値を設けて全過去データからプロジェクトを選定する．例えば，開発期間について 12ヶ月を閾値とした場合，見積り対象プロジェクトの開発期間が 10ヶ月であったならば，開発期間 \leq 12ヶ月を満たすプロジェクトを全過去データから選定し，フィットデータ候補とする．また，見積り対象プロジェクトの開発期間が 15ヶ月であったならば，開発期間 $>$ 12ヶ月を満たすプロジェクトを選定する．手順の詳細を付録 1 に示す．ケーススタディ (3章4節) では，比尺度変数の閾値には中央値を用いた．今回用いたデータセットの比尺度変数は分布に偏りがあるため，平均値よりも中央値がより適していると判断した．

4. ケーススタディ

4.1 概要

ケーススタディの目的は，現実のソフトウェア開発プロジェクトの定量的データを用いて提案方法の有効性を評価することである．実験では，ISBSG[14] が収集したソフトウェア開発企業の実績データを用い，要件定義および設計が終了した時点で総工数を見積る状況を想定した．また，実験時間の都合上，フィットデータの選定において用いる特性の組み合わせの数を 2 つまでに限定した．提案方法のステップ 1 における最小フィットデータサイズを 30 に設定し，選定されたフィッ

トデータ候補に含まれるプロジェクトが 30 件に満たない場合にはその候補を破棄するものとした。これは母集団からサンプルを抽出する場合、30 件が統計上のひとつの目安となるためである [27]。ただし、本実験の主目的は、フィットデータ選定方法による見積り精度の向上の検証であり、最小フィットデータサイズを 30 件とすることが最適であることを保障するものではないことに注意されたい。

ステップ 2 における各モデルのフィットデータ候補に対する適合度を表す指標としては、残差平方平均 (RMS; Residual Mean Square)、自由度調整済み決定係数 (Adj.R²; Adjusted R-square)、最小絶対偏差 (LAD; Least Absolute Deviation) [20]、赤池情報規準 (AIC; Akaike Information Criterion) [1] などが知られている。しかしながら LAD、AIC については、データセットのサイズが値に影響を与えるため、フィットデータのサイズが変化する提案方法では、これらの指標を用いると提案方法の性能を正しく評価できない可能性がある。このため本実験では、重回帰分析における数式モデルの算出仮定において重要な指標である、RMS、Adj.R² の 2 種類を用いた。また提案方法 (3 章 3 節) のステップ 2 におけるモデル構築には、ステップワイズ重回帰分析を用いた。

提案方法との比較対象として、フィットデータの選定を行わない、従来のステップワイズ重回帰分析、およびプロジェクト間の類似度を用いた代表的見積り方法である Analogy-based Estimation (アナロジーベース法)[4]、開発形態によって適用するモデルを変更する開発形態 2 分法を用いた。アナロジーベース法における変数選択法としては、Correlation Threshold based Selection (CTS)[46] を採用した。開発形態 2 分法は、フィットデータを新規開発プロジェクトと拡張開発 (改造、再利用、バージョンアップなど) プロジェクトに分割してフィットデータを作成し、新規開発プロジェクトを見積る際には新規開発プロジェクトだけのフィットデータから構築したモデルを、拡張開発プロジェクトには拡張開発プロジェクトだけのフィットデータから構築したモデルを適用する方法である。この方法は、プロジェクト特性に応じたモデルを人手により構築する一例として位置づけられる。

4.2 アナロジーベース法による工数見積り

提案方法と同様に、見積り対象プロジェクトごとに過去データを選定し、見積りに用いる方法として、アナロジーベース法がある。アナロジーベース法では、「プロジェクト特性（説明変数）が似ているプロジェクトはその工数（目的変数）も似ている」という仮定に基づき、過去プロジェクトと見積り対象プロジェクトとの間の類似度を算出し、類似度の高いプロジェクトのみを用いて見積りを行う。アナロジーベース法以外に、Case-based Reasoning 法 [3][16]、Collaborative Filtering based Estimation 法 [28][38] などが存在するが、類似度計算や見積り値計算のアルゴリズムが異なるだけで基本的な考え方は同じである。

提案方法もアナロジーベース法も、「見積り対象プロジェクトに合わせた過去データの選定を行う」という点では同種のものであると見なせることから、本実験では、提案方法とアナロジーベース法の性能比較を行う。提案方法とアナロジーベース法の大きな違いは、提案方法はモデルベース推論（見積りモデル構築）のためのフィットデータ選定法であるのに対し、アナロジーベース法はメモリベース推論に分類され、モデルが構築されない点が異なる。アナロジーベース法は、近年有望視されている方法ではあるが、変数選択と変数の重み付け方法が確立されていない、類似度計算や見積り値計算に多くのバリエーションがある、公開されているツールが限られているなど、容易に利用できないという課題がある。本章では、アナロジーベース法に基づく見積りツールとして最もよく用いられている ArchAngel[35] で採用されている方法を用いた。また、変数選択法としては、CTS を用いた。ArchAngel の工数見積りの手順は次の通りである。ここでは見積り値算出に用いる類似ケース数を k とおく。プロジェクト類似性に基づく工数見積りでは、プロジェクト間の類似度の算出に先立って、まずデータに含まれる値の範囲を $[0,1]$ に正規化する。この正規化は値域のばらつきによる変数ごとの類似度に対する影響度合いの差を排除するために行われる。 i 番目のプロジェクトの項目 j における変数の値を $v_{i,j}$ 、 j 番目の変数の最大値を $\max(P_j)$ 、最小値を $\min(P_j)$ としたとき、 $v_{i,j}$ の正規化の値 $v'_{i,j}$ は式 (3.1) によって求めるのが一般的である。

$$v'_{i,j} = \frac{v_{i,j} - \min(P_j)}{\max(P_j) - \min(P_j)} \quad (3.1)$$

次に、見積り対象のプロジェクトと過去に行われたプロジェクトとの類似度を、すでに収集された変数の値から計算する。類似度計算の方法として、特性値の系列をベクトルとみなし、ベクトル間の角度を用いる方法や、ベクトル間のユークリッド距離 (Euclidian Similarity) を用いる方法が提案されている。いずれの方法においても変数の値が似ているほど、より類似したプロジェクトであるとみなされるが、欠損値 (データセットに含まれる未記録の値) の取り扱い方に違いがある。前者は欠損値を含んだまま類似度を計算できるのに対し、後者は欠損値を含まない実績データが前提となる。本論文では、欠損値の存在が結果に影響することを避けるため、欠損値のないデータおよびユークリッド距離 [33] を用いる方法を採用した。プロジェクト p_a とプロジェクト p_i の間のユークリッド距離 $dist(p_a, p_i)$ は式 (3.2) によって求められる。

$$dist(p_a, p_i) = \sqrt{\sum_j (v'_{a,j} - v'_{i,j})^2} \quad (3.2)$$

最後に類似度が高いプロジェクトの工数の加重平均により、見積り対象プロジェクトの工数の算出 (見積り値計算) を行う。この際に見積り対象プロジェクトよりも規模が大きい、または小さいプロジェクトが類似度の高いプロジェクトとして多数選ばれた場合、それらの工数の単純な平均では見積りが外れる可能性が高くなる。そこで、見積り値計算時にプロジェクトの規模を補正する $amp(p_a, p_i)$ を重みとする加重平均をとることにする。ここで $amp(p_a, p_i)$ を式 (3.3) のように定義し、 i 番目のプロジェクトの実績工数を $effort_i$ 、 i 番目のプロジェクトの正規化後の開発規模を FP'_i とすると、見積り対象である a 番目のプロジェクトの見積り値 $effort'_a$ は式 (3.4) のように定義される。

$$amp(p_a, p_i) = \frac{FP'_a}{FP'_i} \quad (3.3)$$

$$effort'_a = \frac{\sum_{i \in k\text{-nearestprojects}} (effort_i \times amp(p_a, p_i))}{k} \quad (3.4)$$

ここで k -nearest projects は、プロジェクト p_a との類似度が高い k 個のプロジェクトの集合を表す。

4.3 利用したデータセット

実験には ISBSG が収集しているデータセットを用いた。採用の理由は 2 章 2 節で述べたのと同様である。サブセット構築の手順についてもおおむね同様であるが、細部について違いがあるため、重複を含めて記述する。

本実験では、ISBSG のデータセットから、文献 [15]、文献 [24] にならってデータクオリティレーティングが C または D のプロジェクトを除外した。次に FP 数の測定方法が IFPUG ではないものを除外した。これらの処理を行った後に、例えば計画工数などの工数に関わる特性などの、致命的な欠損を含むプロジェクトや欠損しているプロジェクトが多い特性を除外し、欠損を含まない 232 件のプロジェクト、8 種類の特性を含むデータセットを作成し、実験に用いた。3 章 4 節 1 項で述べたとおり、本実験では設計終了時を想定しているため、この 8 種類の特性は全て設計終了時まで決定可能と考えられる変数のみを採用している。ここで構築されたサブセットが 2 章 3 節のものと異なっているのは、本実験では選定後のデータセットに含まれるプロジェクトが 30 件以下の場合、そのデータセットを除外するためである。選定の対象となるデータセットに含まれるプロジェクト数が少ない場合、大半のプロジェクトが 30 件以下となり、選定の効果が表れないおそれがある。このことから、本実験では特性の数よりもプロジェクト数を優先してサブセットの構築を行った。

8 種類のプロジェクトの特性について、特性値名、尺度の種別、名義尺度に含まれるカテゴリ名の一部、それぞれの名義尺度に含まれる特性値の種類数を表 3.1 に示す。特性の詳細については文献 [10] を参照されたい。8 種類の特性変数には 4 種類の比尺度変数、3 種類の名義尺度変数、1 種類の順序尺度変数が混在している。このうち名義尺度変数および順序尺度変数については、それぞれ 0 または 1 の値をとる複数の 2 値変数（ダミー変数）に変換したものをを用いた。

実験では、データセットを無作為に半数ずつに分け、一方をフィットデータ、も

表 3.1 実験用 ISBSG データセットに含まれるプロジェクトの特性

特性値名	尺度	含まれる特性	特性値数
FP 数	比尺度	数量データ	—
総工数	比尺度	数量データ	—
計画工数	比尺度	数量データ	—
要件定義工数	比尺度	数量データ	—
開発形態	名義尺度	新規開発, 拡張開発, 再開発	3
開発言語	名義尺度	COBOL, Java, C 等	24
工数計測方法	名義尺度	実作業時間集計, 報告作業時間集計 等	7
工数算入業務	順序尺度	レベル 1~4	4

う一方をテストデータとした。この操作を独立に 30 回繰り返し、フィットデータとテストデータのペアを 30 組作成した。各データの組を用いて、目的変数である総工数の見積りを行った。

4.4 精度評価指標

実験では、見積り誤差を評価する基準として、実績値を分母に置く相対誤差 MRE (Magnitude of Relative Error), および見積り値を分母に置く相対誤差 MER (Magnitude of Error Relative)[17] の 2 つを用いる。MRE が高ければ、見積りが過大見積りであったことを示し、MER が高ければ、見積りが過小見積りであったことを示す。それ以外の誤差指標としては絶対誤差もよく用いられているが、本章では、MRE と MER の両方を用いることで、過小見積り、過大見積りの双方を評価できることから、絶対誤差を用いての評価は割愛する。

見積り方法の評価基準として、MRE と MER, それぞれの平均値 (MMRE, MMER), 中央値 (MdMRE, MdMER), それらの値が 0.25 以下のプロジェクトが占める割合 (Pred(25).MRE, Pred(25).MER) の 3 種 6 個を用いる。Pred (Percentage of prediction) とは、相対誤差が特定の値以下のプロジェクトが全体

に占める割合を示すもので，Pred(25).MRE は相対誤差 MRE が 25%以下のプロジェクトが全プロジェクト数に占める割合を表す．従って，Pred(25).MRE が大きいほど MRE での見積り精度が高いことを示す．Pred(25).MER についても同様である．以下に MRE (再掲) と MER，Pred(25) の計算式を示す．式中，実績値を E で，モデルから算出される値 (見積り値) を \hat{E} で表し， M は見積り対象プロジェクト全体の件数を表す．また，式 3.6 は Pred(25).MRE の算出式であり，Pred(25).MER は式中の MRE を MER に置換したものである．

MRE 計算式

$$MRE = \frac{|E - \hat{E}|}{E}$$

MER 計算式

$$MER = \frac{|E - \hat{E}|}{\hat{E}} \quad (3.5)$$

Pred(25).MRE 計算式

$$Pred(25).MRE = \frac{\sum_{i=1}^M isAccurate(MRE_i)}{M} \quad (3.6)$$

$$isAccurate(MRE) = \begin{cases} 1 & MRE \leq 0.25 \\ 0 & MRE > 0.25 \end{cases}$$

4.5 モデル適合度と精度の関係の調査

4.5.1 調査概要

本実験では RMS と Adj.R² の 2 種類をモデル適合度として用いることとした．しかしながら，提案方法が有効であるためには，2 種類の指標のうちどちらか 1 つは見積り精度との間に関係 (相関) がなくてはならない．もし何の関係がない

のであれば、その適合度を用いてモデルを選択しても精度の向上は見込めないからである。そこで、本実験を行う前に、RMS と Adj.R^2 の 2 種類の指標と見積り精度の相関について調査実験を行うこととした。ただし、簡便のために評価指標としては MRE のみを用いることとした。

調査実験では、本実験と同様にデータセットをフィットデータとテストデータに無作為に 2 分割する、という作業を 30 回繰り返し、各データの組を用いて調査を行った。

4.5.2 調査手順

調査実験では、まず 3 章 3 節で説明した提案方法のステップ 2. までを行い、層別後のフィットデータおののからモデルを構築する。次に、構築されたモデル全てを予測対象のプロジェクトに適用し、それぞれ見積り値を得る。これによって、各モデルは構築時に適合度が得られ、見積り値からは相対誤差を算出できるため、適合度と相対誤差のセットが各特性について得られる。

調査実験では、各プロジェクトについて、そこから構築可能なモデルの適合度と、モデル適用時の相対誤差の間の相関係数をスピアマンの順位相関を用いて算出し、1 回の試行で得られる全相関係数の中央値、平均値を、そのプロジェクトにおける相関係数として採用した。

4.5.3 調査結果

表 3.2 と図 3.2 に、30 回の試行結果のまとめを示す。表 3.2 中の中央値とは、全試行から得られた相関係数 (116 件 \times 30 回の試行 = 3480 件) の中央値である。また図 3.2 は相関係数を箱ひげ図で示したものである。

さらに、予測対象プロジェクトに適用可能なモデルの中で適合度が最も高いモデルを適用したときの、相対誤差の順位を調査したものを図 3.3、図 3.4 に示す。横軸は相対誤差順位を、左側縦軸はその相対誤差順位のプロジェクトの件数を、右側縦軸は、相対誤差順位がそれ以上のプロジェクトが全体の何パーセントを占めるかを示している。またこれらの図では、表 3.2 のように試行ごとに中央値を

表 3.2 モデル選択指標ごとのモデル適合度と相対誤差の相関係数

	中央値	平均値
提案方法 RMS 優先	0.284	0.206
提案方法 Adj.R ² 優先	-0.012	0.016

取るなどの処理をせず，30 回の試行に含まれる結果を全て示している．ここで「相対誤差の順位」について説明する．3 章 4 節 5 項 2 において述べた手順によって，予測対象プロジェクトに適用可能な各モデルの適合度と，それに対応する相対誤差が得られる．次に，モデルの適合度群と相対誤差群についてそれぞれ順位付けを行う．これは本章で提案する方法では，適合度が最も良い，すなわち適合度の順位が 1 位のモデルを予測対象プロジェクトに適用するため，その場合の相対誤差順位が重要となるためである．

表 3.2，図 3.2 から，中央値，平均値ともに RMS の場合には適合度と相対誤差にはある程度の相関があり，Adj.R² ではほとんど相関がないという結果が得られた．図 3.3，図 3.4 から，RMS を採用した方が精度が高くなりやすい傾向が見られた．さらに（層別を行わずに）全てのフィットデータを用いて構築したモデルを適用した時の相対誤差順位の平均値が 9.3 であるのに対して，RMS が 1 位のモデルを適用したときの相対誤差順位の平均値は 7.4，Adj.R² が 1 位のモデルを適用したときの平均値は 8.2 であった．このことから提案方法による精度の向上が期待できることが明らかとなり，特に，RMS を選定に用いることで，より高い精度での見積りが期待できる．

これらの結果から，少なくとも RMS には見積り精度との間に相関があり，提案方法において適合度を用いることの妥当性が確認できた．ただし，Adj.R² は重回帰分析において変数の投入を決定するための重要な指標であるため，本実験でも一応，採用しておくことにする．

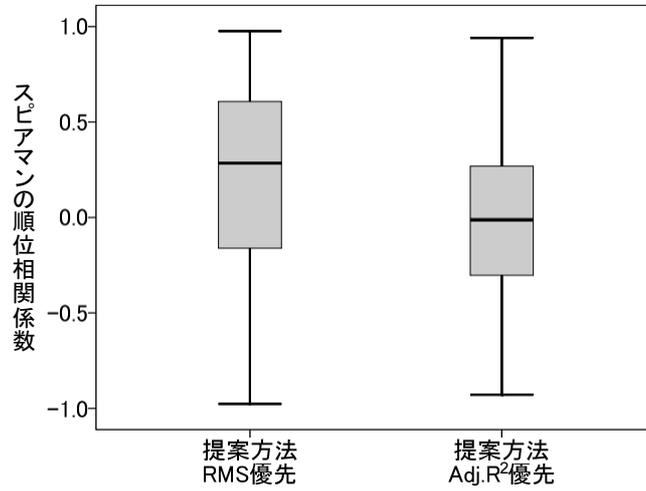


図 3.2 適合度指標ごとの相関係数の箱ひげ図

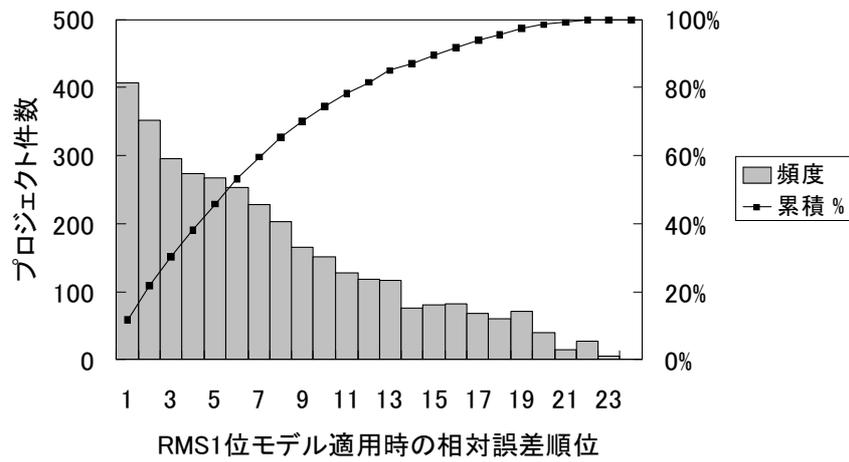


図 3.3 RMS 順位 1 位モデル適用時の相対誤差順位のヒストグラムとその累積%

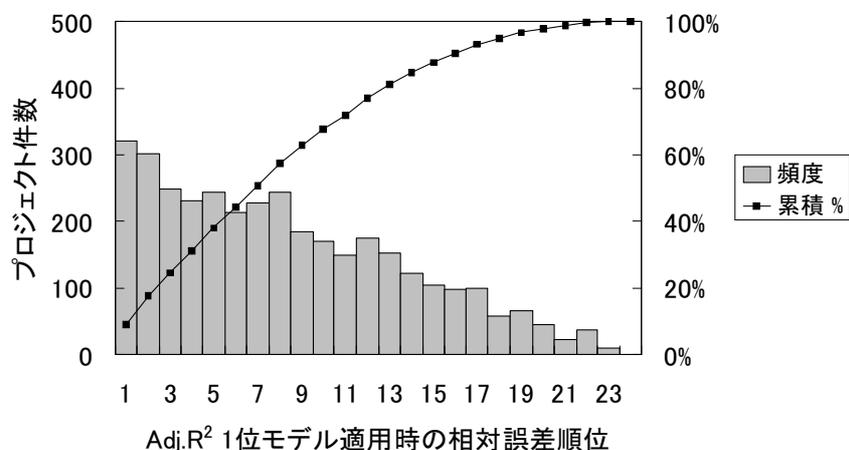


図 3.4 Adj.R² 順位 1 位モデル適用時の相対誤差順位のヒストグラムとその累積 %

4.6 結果

提案方法を用いての見積り結果を，MRE は表 3.3 および図 3.5 に，MER は表 3.4 および図 3.6 に示す．図 3.5，図 3.6 では，30 回の試行のそれぞれで得られた MdMRE および MdMER の分布を箱ひげ図で示している．表中および図中では，全てのデータセットを用いてモデルを構築したものを「従来法」と表記し，提案方法において適合度を表す指標に残差平方平均を用いたものを「提案方法 RMS 優先」，自由度調整済み決定係数を用いたものを「提案方法 Adj.R² 優先」，アナロジーベース法を用いたものを「アナロジーベース法」と表記した．また表中の Pred(25) は 30 回の試行それぞれで得られた評価基準を平均したものであり，平均値，中央値および分散については 30 回の試行の結果の中央値を用いた．

まず，提案方法において RMS 優先と Adj.R² 優先を比較すると，前者の方が全ての評価基準において高い精度を示した．また Wilcoxon 検定を行った結果，MdMRE では $p = 0.00171$ ，MdMER では $p = 0.000975$ と，どちらの尺度の場合でも有意水準 1% で RMS 優先の精度が高いことが確認できた．そこで，以降では，提案方法の RMS 優先と他の方法との比較を行う．

提案方法（RMS 優先）と従来方法を比較すると，提案方法は MRE では全て

表 3.3 提案方法とそれ以外の方法による見積り精度の比較 (MRE)

	提案方法 RMS 優先	提案方法 Adj.R ² 優先	従来法	アナロジー ベース法	開発形態 2分法
MMRE	0.640	0.761	0.782	0.592	0.708
MdMRE	0.367	0.397	0.452	0.410	0.436
Pred(25).MRE	35.5	32.0	31.7	31.8	31.8

表 3.4 提案方法とそれ以外の方法による見積り精度の比較 (MER)

	提案方法 RMS 優先	提案方法 Adj.R ² 優先	従来法	アナロジー ベース法	開発形態 2分法
MMER	0.592	0.709	0.507	0.595	0.536
MdMER	0.336	0.358	0.357	0.385	0.360
Pred(25).MER	38.7	34.8	34.7	33.2	34.0

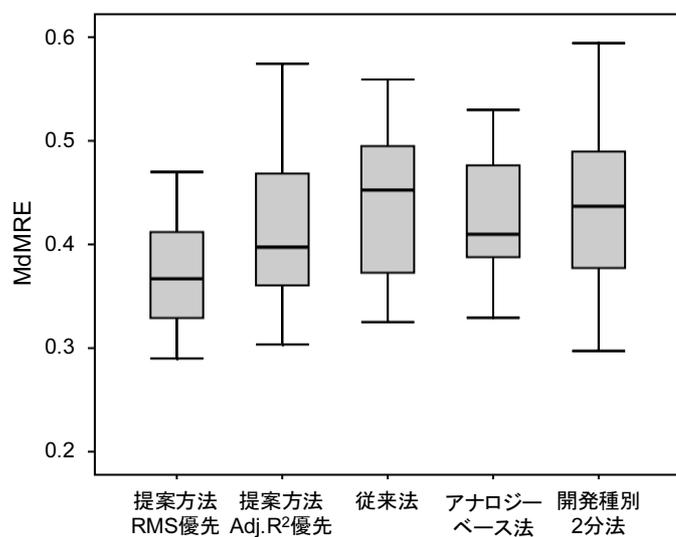


図 3.5 提案方法とそれ以外の方法による見積り精度の箱ひげ図 (MRE)

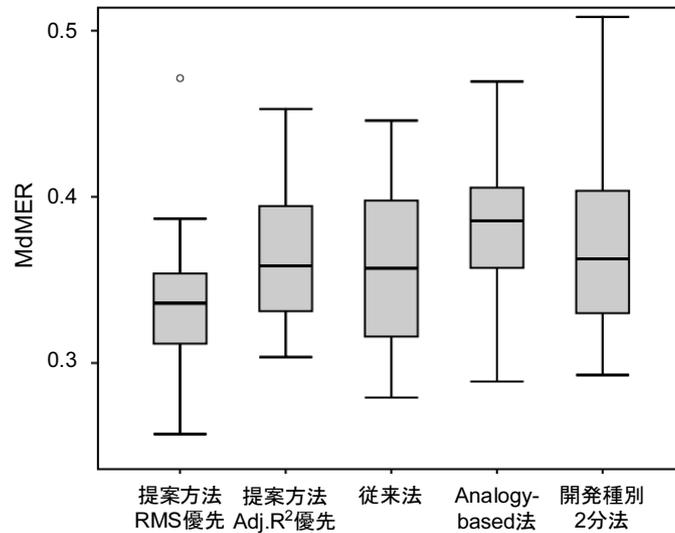


図 3.6 提案方法とそれ以外の方法による見積り精度の箱ひげ図 (MER)

の指標において従来方法からの精度の改善が見られた。しかしながら MER では MdMER, Pred(25).MER では精度の改善がみられたものの, MMER の精度は低下していた。アナロジーベース法との比較では, MER では全ての評価基準において提案方法が優れていたが, MMRE ではアナロジーベース法の精度が優れているという結果になった。開発形態 2 分法との比較では, 従来方法との比較と同様の結果となり, 提案方法が常に優れているとは言えない。しかしながら, どの場合にも 6 種類の評価基準のうち 5 種類で提案方法の方が優れており, このような場合, 過去の工数見積りに関する論文では, より多くの評価基準で勝っている方を優れているとするのが一般的であり [20] [25] [30], 本章でも従来方法, アナロジーベース法, 開発形態 2 分法よりも提案方法の方が優れていると考えられる。Wilcoxon 検定を行った結果, 提案方法と従来方法の組み合わせでは MdMRE は $p = 0.000$, MdMER は $p = 0.006$, 提案方法とアナロジーベース法との組み合わせでは MdMRE は $p = 0.009$, MdMER は $p = 0.000$ と, 有意水準 1% で提案方法の精度が高いことが示された。また提案方法と開発形態 2 分法の組み合わせでは MdMRE は $p = 0.015$, MdMER は $p = 0.013$ と, 有意水準 5% で提案方法の精度が高いことが示された。

4.7 考察

まず，提案方法に用いる適合度の指標として RMS 優先を用いた場合と Adj.R^2 優先を用いた場合を比較すると，前者の方が高い精度を示した．この結果から，提案方法に用いる適合度を表す指標としては，RMS を使うことを推奨する．このような結果となった一つの解釈は，RMS 優先では推定値と実績値との残差が（平均的に）小さいモデルが選択されるのに対し， Adj.R^2 優先では推定値と実績値との相関係数が大きいモデルが選択されるが，相関係数は外れ値に影響を受けやすいため， Adj.R^2 優先の精度が低下した可能性がある．

以降では，RMS 優先と Adj.R^2 優先の違いの詳細を明らかにするために，各プロジェクト特性について，フィットデータ選定基準として採用された回数を調査した結果を表 4 に示す．表 3.5 中の値は，それぞれのフィットデータが最終的に選択された回数が全体に占める割合を示した．ただし，RMS 優先および Adj.R^2 優先において選択された割合がどちらも 5% に満たないものについては省略している．

表 3.5 から，RMS 優先では，開発規模，システム化計画工数，要件定義工数などの比尺度の特性，特にそれが下位（中央値よりも小さい）のプロジェクトがフィットデータとして選定される傾向にあり，特に，計画工数が下位のプロジェクトのみを選定してフィットデータとするケースが多かった．一方， Adj.R^2 優先では，開発形態や開発言語などの名義尺度の特性，そして全ての過去プロジェクトをフィットデータとするケースが多くみられた．実験結果において RMS 優先が Adj.R^2 優先よりも高い精度で見積りができたことを考えると，規模が小さいプロジェクトの見積りでは，開発形態などの名義尺度の変数よりも，規模や工数などの比尺度の変数が共通するプロジェクトを選定し，フィットデータとしたほうが，高い精度で見積りが可能だと考えられる．

RMS 優先， Adj.R^2 優先に共通する特徴としては，開発形態が拡張開発の場合にはその特性を含むプロジェクトがフィットデータとして選択されているのに対して，新規開発の場合には，開発形態が新規開発かつ，それ以外の特性を持つという 2 つの条件を満たすプロジェクトがフィットデータとして選択されている．このことは，新規開発プロジェクトの総工数を見積る場合には，過去プロジェクト

表 3.5 特性ごとのモデル選択割合

特性変数 1	特性値 1	特性変数 2	特性値 2	RMS 優先	Adj.R ² 優先
全過去データ	—	—	—	7.5	13.1
FP 数	下位	—	—	7.8	1.8
計画工数	下位	—	—	16.1	2.0
要件定義工数	下位	—	—	11.0	0
開発形態	拡張開発	—	—	5.1	22.2
開発言語	COBOL	—	—	1.9	8.3
FP 数	上位	開発形態	新規開発	3.1	7.3
FP 数	下位	開発形態	新規開発	2.9	6.0
要件定義工数	下位	開発形態	新規開発	7.3	1.3
要件定義工数	下位	工数算入業務	レベル 1	8.0	2.5

から新規開発プロジェクトを抽出し、さらに FP 数や計画工数、要件定義工数などの総工数と密接に関わりのある特性によって選定を行うことが精度を高める可能性があることを示唆している。

また特筆すべきこととして、RMS 優先、Adj.R² 優先の双方において全ての過去データをフィットデータしたものが少なからず選択されている。このことは、フィットデータを特性によって選択することが常に精度を高めるわけではないことを示している。一般に特性に基づくフィットデータの選定は見積り精度を高めると考えられがちであるが、この結果は見積り対象のプロジェクトによっては、選定を行わずに全ての過去のプロジェクトをフィットデータとして扱った方が精度が高くなる可能性を示唆しており興味深い。見積り精度を高めるためには、フィットデータの選定において全ての過去のプロジェクトをフィットデータ候補の一つとして含めることが重要であると考えられる。

次に提案方法（RMS 優先）と従来方法を比較した場合に、MER の平均値である MMER において提案方法が低い精度を示した原因について考察を行う。考えられる原因として、提案方法では、見積り結果に外れ値が存在したことが挙げられる。平均値は中央値よりも外れ値に影響を受けやすいことから、MdMER では高い精度を示していても、MMER では低い精度となったと考えられる。現実

のデータセットには，多くの場合外れ値となるようなプロジェクトが含まれており，提案方法ではそのようなプロジェクトをうまく見積りできていない可能性がある．しかし，MdMER では高い精度を示していることから，外れ値とならない多くのプロジェクトをより高精度で見積りできており，提案方法の有用性は高いと考える．

5. まとめ

本章では過去のソフトウェア開発プロジェクトの実績データを用いて，重回帰分析により開発工数見積りモデルを構築するという状況において，見積り対象プロジェクトの特性に基づいて，モデル構築のためのフィットデータに適したプロジェクト群を選定する方法を提案した．フィットデータの選択を行わない（全過去プロジェクトを用いる）従来の重回帰分析と比較すると，相対誤差 MRE の中央値では 0.452 から 0.367 へ，相対誤差 MER の中央値では 0.357 から 0.336 へと精度が向上した．またアナロジーベースと比較した場合にも，多くの評価指標で精度が高いことが示された．

本章での実験結果から，見積り対象の特性に合わせてフィットデータを選定することは重要であり，それを系統的に行う方法の 1 つとして，提案方法が役立つことが示された．同時に，フィットデータの選択結果から選定時の候補の中に全ての過去データを用いるケースを含めることもまた，精度を高める上では重要であることが示された．

実際のソフトウェア開発における提案方法の適用可能性としては，提案方法は 1 つ以上のプロジェクト特性が記録されていれば利用できるもので，原理上はソフトウェア開発のどの時点においても適用可能である．特に高い精度での見積りが可能となる業務要件設計終了以降 [41] であれば，開発プロセスの進行に伴い利用可能な特性の種類は増加する．開発プロセスの節目節目において提案方法を適用することで，プロジェクトの特性により即したフィットデータとモデルが選択可能となり，精度の向上が期待できる．ソフトウェア開発企業内で提案方法を適用する場合に精度をさらに向上させる方法としては，ケーススタディでは 30 件と

したモデル構築のための最小フィットデータ数を、企業で蓄積されているデータに合わせて適切に変更することが考えられる。それに加えて、フィットデータに含まれる特性を結合する、例えば開発言語のCとC++を1つのものとして扱う、といった作業を適切に行うことでも、精度の向上が可能であると考えられる。提案方法の適用結果のソフトウェア開発へのフィードバック方法としては、まず、開発初期の時点で見積りを行い、見積り結果を開発現場に示し、納期と見積り工数から調達すべき人員の数を決定することが考えられる。そして、開発が進むとともに繰り返し見積りを行うことで、見積り工数と実際の工数の乖離から、進行中のプロジェクトの状況を検証でき、問題の発生の把握に役立つ。また、人手による工数見積りが併用されている場合には、モデルによる見積り結果と比較することで、人手による見積りの確からしさを確認したり、見積りの再検討をする場合にも有用である。さらに、見積り工数とともに、選定されたフィットデータ集合および対象プロジェクトと一致した特性値を開発現場に提供することで、見積りの根拠の確認が可能となり、見積り結果が開発現場に受け入れられやすくなると期待される。

なお、現代のソフトウェア開発は拡張開発が多く、そのような開発プロジェクトでは提案方法による工数見積りに先立って母体システムの調査を行い、母体の変更箇所や規模を明確にしておくことが望ましい。開発の契約の観点からは、母体の調査を含む上流工程を準委任契約として開発規模の見積りを行い、改めて請負契約を結ぶことが奨励される [39]。

提案方法はモデルベースの見積り方法であればニューラルネットなどにも適用可能である。また、2クラス判別問題のための線形判別モデルやロジスティック回帰モデルなどにも応用可能である。今後、重回帰分析以外の見積り方法に対する提案方法の有効性評価が課題となる。

第4章 ソフトウェア開発工数見積り における見積り回避プロジェクトの選定とその効果

1. まえがき

ソフトウェア開発において、工数の見積りはプロジェクトを予定通りに完遂させるために必要な人的資源の決定やスケジュール管理を行う上で重要である。このために高い精度で工数を見積ることを目的として数多くの研究が行われてきた [4] [33] [34] [38]。

図 4.1 に筆者らが以前に行った工数見積りの結果を再掲する [43]。ISBSG のデータセットから作成した、欠損のないサブセットに対してステップワイズ重回帰モデルを用いて見積りを行ったものである。ISBSG は実際のソフトウェア開発プロジェクトで得られる工数や開発言語などのプロジェクトデータの収集を行って

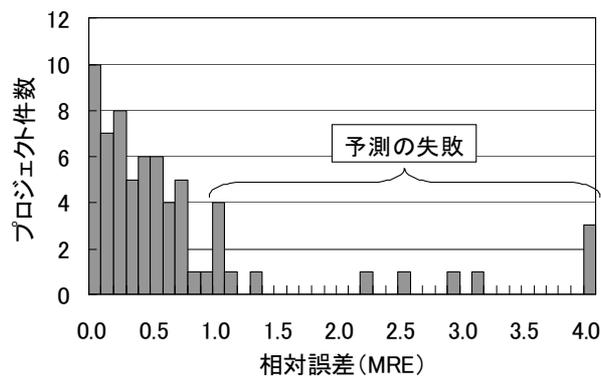


図 4.1 重回帰分析による見積り結果の一例

る組織である。横軸は相対誤差，縦軸はプロジェクトの件数を表す。図から，見積り結果には相対誤差が30%以下のプロジェクトが多数含まれており，これらのプロジェクトでは見積りが成功したといえる。しかし，相対誤差が100%を超える，いわば見積りに失敗したプロジェクトも，成功したプロジェクトほどではないにせよ含まれている。この結果は全体としての見積り精度がどれほど高くとも，いくつかのプロジェクトでは見積りに失敗する可能性があることを示している。そして見積りに失敗する可能性があるならば，見積り結果を得ることができてもその結果を用いることは難しい。これまでの工数見積り方法に関する研究では平均的な精度の向上のみが注目され，大きく見積りが外れる可能性，すなわち見積りの失敗が注目されることはほとんどなかった。

そこで本章では「見積り対象プロジェクトと類似プロジェクト群の間の類似度の分散または平均が大きい場合，その見積りは失敗しやすい」と「見積り対象に類似していると判定された類似プロジェクト群の工数の変動係数と分散が大きい場合，その見積りは失敗しやすい」という2つの仮説を立て，類似するプロジェクトがもつ特徴を利用して見積りに失敗しやすいプロジェクトを選定する。そして，見積りに失敗すると判定されたプロジェクトは見積り自体を回避すべきプロジェクト（見積り回避プロジェクト）として，そもそも見積り対象から外すことを考える。仮説の検証のために，アナロジーベース法において，見積りに失敗したプロジェクトとその類似プロジェクトの特徴を示す値との関係を実際のソフトウェア開発プロジェクトで得られたデータを用いて検証する。見積りに失敗したプロジェクトの定義は，見積りの相対誤差（MRE）が100%を超えるプロジェクトとした。

以降，4章2節では仮説について詳しく述べ，4章3節では仮説を検証する実験の概要とその結果，考察を述べる。最後に4章4節で本章の結論について述べる。

2. 仮説

本章では，アナロジーベース法において2つの仮説を立て，その仮説に基づいて見積りに失敗しやすいプロジェクトを選定，アナロジーベース法による見積り

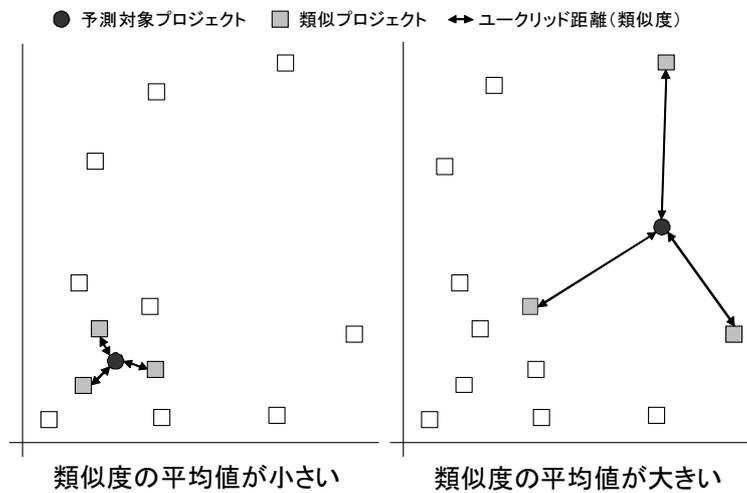


図 4.2 類似度の平均が小さい場合と大きい場合の例

対象から外す方法について検討する。2つの仮説は、見積り対象プロジェクトと類似プロジェクトの間の類似度に着目したものと、見積り対象プロジェクトに類似していると判定されたプロジェクト（類似プロジェクト）の開発工数（総工数）に着目したものから成る。前者を仮説1，後者を仮説2として，以下に詳細を述べる。

仮説の説明では，類似プロジェクト件数は3件とし，類似度として Euclidian Similarity を用いた場合を考える。2つのプロジェクト間の Euclidian Similarity の値が小さいならば，そのプロジェクトは類似していると判定される。この場合，見積り対象のプロジェクトからみて，Euclidian Similarity が最小のプロジェクトから3件が類似プロジェクトとして扱われる。

仮説1 これは「見積り対象プロジェクトと類似プロジェクト群の間の類似度の分散または平均が大きい場合，その見積りは失敗しやすい」という仮説である。類似度の平均を用いた場合の例を図4.2に示す。この仮説の根拠は，アナロジーベース法においては，見積り対象のプロジェクトに似ていると判定されることと，類似プロジェクト群の類似度が高いことが常に両立するとは限らないためである。まず類似度の分散が大きい場合を考える。例えば，3件のうち2件については類似度が十分高いが，その次に近いと判定さ

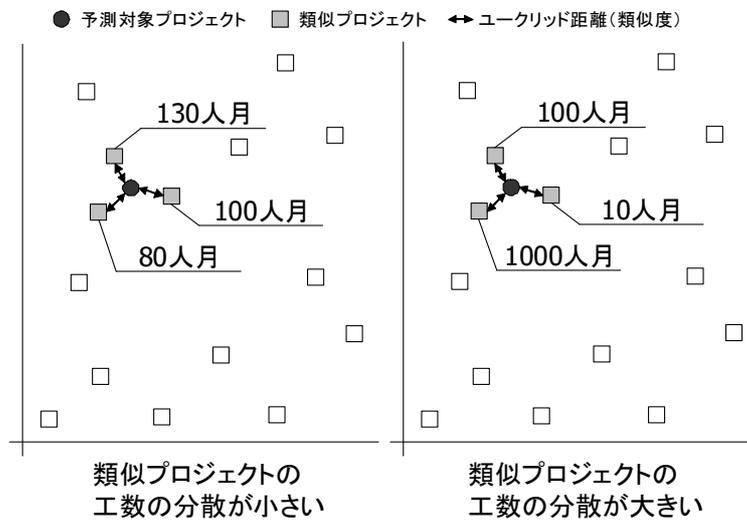


図 4.3 類似プロジェクトの工数が小さい場合と大きい場合の例

れた、3件目のプロジェクトの類似度が非常に低い場合には、本当に類似しているプロジェクトとそうでないプロジェクト混在させて見積りに用いていることになり、見積り精度が低くなる可能性がある。次に類似度の平均が大きい場合、類似プロジェクトと判定されているプロジェクトが、実際にはそれほど見積り対象のプロジェクトには類似していないことを意味する。実際には類似していないプロジェクトを用いて見積りを行っても、その精度は期待できない。

仮説2 これは、「見積り対象に類似していると判定された類似プロジェクト群の工数の変動係数と分散が大きい場合、その見積りは失敗しやすい」という仮説である。変動係数とは標準偏差を平均値で割った値であり、変動係数も分散も値の分布を表す指標である。類似プロジェクトの総工数の分散を用いた場合の例を図4.3に示す。類似性に基づく工数見積りでは、類似プロジェクトの工数を加重平均して見積り工数を得る。その算出過程において、類似プロジェクトの工数のばらつきは考慮されない。そのため、類似プロジェクトの工数の信頼性に問題があったとしても、その値がそのまま見積り工数の算出に用いられる。この類似プロジェクトの問題がある場合の例として、類似プロジェクトの工数の分散または変動係数が大きい場合を考

える。もし、ある見積りプロジェクトについて類似度が非常に高い3件を検出できたとしても、その3件の工数が、それぞれ10人月、100人月、1000人月であった場合、記録ミスや特異プロジェクトを含む可能性が高く、算出される工数は実際のものとはかけ離れた値となるおそれがある。

このように、2つの仮説のどちらが成り立つ場合にも、見積り精度は低くなる可能性がある。4章1節でも述べたとおり、どれほど全体として精度が高くとも、見積りに失敗する可能性があるならば、その結果を採用することは難しい。そのような見積りに失敗しやすいプロジェクトは見積り対象から除外すべきである。それゆえ、仮説が正しいならば、類似プロジェクトがその条件を満たす見積り対象プロジェクトは、アナロジーベース法による見積りを回避すべきであると考えられる。以降、仮説1と仮説2で用いた「類似度の分散または平均」と「類似プロジェクト群の工数の変動係数と分散」を便宜上まとめて「ばらつき」と呼ぶ。

3. ケーススタディ

3.1 概要

ケーススタディの目的は、4章2節で述べた2つの仮説が、現実のソフトウェア開発プロジェクトで得られた定量的データを用いた場合に成立するかを確かめ、類似度の平均値、分散および類似プロジェクトの工数の変動係数、分散による選定が見積りに失敗するプロジェクトを特定可能かを検証することである。

実験では3種類の実際の開発プロジェクトから得られたデータを用いて、要件定義および設計が終了した時点を想定して総工数の見積りを行った。そして見積りの結果として得られる、類似プロジェクト群のばらつきを表す指標と見積り精度の関係についての分析を行った。まず、ばらつきの各指標と見積り精度の相関を調べ、どの指標と精度に関係があるかを調べた。その上で、ばらつきの指標の値によって、見積りに失敗するプロジェクトを判別できるかを調べた。以降では、実験に用いたデータ、精度評価指標について述べ、続いて実験手順について述べる。

3.2 利用したデータセット

本研究では、Desharnais データ [8][21]、Kitchenham データ [18]、ISBSG データ [11] の 3 つのプロジェクトデータセットを用いた。これらのデータセットに共通して含まれているデータ項目には、目的変数である総工数、規模を表す FP 数などがある。以下では、それぞれのデータセットについて、より詳細に述べる。Desharnais データは、Desharnais によって収集されたカナダのソフトウェア開発企業における 80 年代のデータである。データには 77 件のプロジェクトについて、プロジェクトマネージャの経験年数やチームの経験年数などの 11 種類のプロジェクト特性が記録されている。また、このデータには欠損値は含まれていない。ただし、このデータセットには開発の終了年などの工数見積りに用いるには不適切な説明変数が含まれていたため、8 種類の特性を含むサブセットを作成し、実験に用いた。

Kitchenham データは、Kitchenham が 2002 年に文献中で公開したソフトウェア開発企業のデータである。データには 145 件のプロジェクトについて、9 種類の変数が記録されている。しかし、このデータセットにも Desharnais データと同様に、工数見積りに用いるには不適切な説明変数や欠損値が含まれていたため、プロジェクト数 135 件、顧客名（匿名化のために番号に置き換えられている）やプロジェクトタイプ（開発 (development)、適応保守 (adaptive)、是正保守 (corrective) 等）など 7 種類のプロジェクト特性を含むサブセットを作成し、実験に用いた。

ISBSG データは、International Software Benchmark Standards Group (ISBSG) が収集した 20 カ国のソフトウェア開発企業の実績データである。データには 1989 年から 2004 年までの 3026 件のプロジェクトについて、それぞれ 99 種類の変数が記録されている。ただし、このデータには数多くの欠損値が含まれているため、欠損を含むプロジェクト、変数を削除し、プロジェクト数 232 件、開発形態や開発言語を含む変数 8 種類の欠損値のないサブセットを作成し、実験に用いた。ここで用いたデータは 3 章で作成したものと同一であるため、プロジェクト特性の表については省略する。

これら 3 つのデータセットでは、説明変数に名義・順序変数と比尺度変数が混在しており、工数を見積る前に、名義・順序変数をダミー変数化（2 値変数化）し、

比尺度変数に変換する必要がある。比尺度変数とは開発規模や開発期間などの、連続値として表現される変数であり、名義・順序変数とは開発言語などがどのカテゴリ（C言語，COBOL，Java など）に属するかという情報が不連続値として表現される変数である。ダミー変数化するには、変数に含まれる各カテゴリに対してそのカテゴリに属するかを表現する2値の変数に変換する。ある名義・順序変数に p 個のカテゴリがあるなら、その名義・順序変数は $p-1$ 個のダミー変数になる。

上記の3つのデータに対してこの変換を行った結果、Desharnais データの変数は8個、Kitchenham データの変数は15個、ISBSG データの変数は41個となった。

実験においてステップワイズ法に代表される、計算による変数選択は行わなかった。

3.3 精度評価指標

ケーススタディでは、見積り誤差として相対誤差 MRE を採用し、ここから導くことのできる MdMRE および $\text{Pred}(\overline{100})$ を見積りの失敗を評価する指標として用いる。MdMRE は MRE の中央値を表し、 $\text{Pred}(\overline{100})$ は相対誤差が 100% を越えるプロジェクトが全体に占める割合を表す。MRE の算出式を再掲（式 2.1）し、 $\text{Pred}(\overline{100})$ の算出式を 4.1 に示す。式中、実績値を E で、モデルから算出される見積り値を \hat{E} で表し、 M は見積り対象プロジェクトの全体の件数を表す。

MRE 計算式

$$MRE = \frac{|E - \hat{E}|}{E}$$

$\text{Pred}(\overline{100})$

$$\text{Pred}(\overline{100}) = \frac{\sum_{i=1}^M \text{isAccurate}(R_i)}{M} \quad (4.1)$$

$$isAccurate(MRE) = \begin{cases} 1 & MRE \geq 1 \\ 0 & MRE < 1 \end{cases}$$

3.4 手順

実験では各データセットに対してリーブワンアウト法を適用し、見積りを行った。見積り値を算出するまでの手順とそれに続く相関分析、相対誤差とばらつきの指標の関係分析手順以下に示す。

見積り値算出手順

手順1 データ中のプロジェクトを1つ選ぶ。

手順2 選んだプロジェクトの実績工数値を隠し、見積り対象のデータ（テストデータ）とする。それ以外のプロジェクトを見積りの根拠として使用するデータ（フィットデータ）とする。

手順3 フィットデータを用いて、テストデータの工数をアナロジーベース法で見積る。類似ケース数は5件とした。

手順4 手順1~4をデータセットに含まれる全プロジェクトに対して行う。

相関分析手順

手順1 各プロジェクトについて算出した工数の見積り値と実績値から MRE を算出する。

手順2 MRE とばらつきの各指標との相関係数を算出し、どの指標と MRE に関係があるのかを調べる。相関係数としてはスピアマンの順位相関係数を用いる。同時に無相関検定を行い、相関が有意か否かについても調べる。

相対誤差とばらつき指標の関係分析手順

手順1 MRE を用いて $\text{Pred}(\overline{100})$ を算出する。

手順2 類似度計算の結果，類似していると判定されたプロジェクト（以降類似プロジェクトと呼ぶ）の実績値からばらつきの各指標を算出する．

手順3 ばらつきの指標と $\text{Pred}(\overline{100})$ の関係をグラフから調査する．

本実験では類似度の算出には Euclidian Similarity[33] と Cosine Similarity[28] を採用した．ばらつきの各指標とは，前述の通り類似度の分散と平均，および類似プロジェクト群の総工数の変動係数と分散である．本実験の場合，変動係数は5件の類似プロジェクトの総工数の標準偏差を平均値で割ったものとなる．ただし規模の違いによる影響を少しでも軽減するために，類似プロジェクト群の総工数は式 3.3 で示した規模による補正後の値を用いた．

相関分析手順の2.においてノンパラメトリックな方法であるスピアマンの順位相関係数を用いたのは，相対誤差に外れ値ともとれる大きな値が含まれており，ピアソンの積分相関係数のようなパラメトリックな方法では相関係数が外れ値の影響を受けやすいためである．

3.5 結果

3.5.1 ばらつきの指標と MRE の相関係数

相対誤差である MRE と，ばらつきの指標である，類似プロジェクトの総工数の変動係数と分散，類似度（Euclidian Similarity と Cosine Similarity）の分散と平均との間の相関係数を表 4.1，表 4.2 に示す．また無相関検定の結果も併せて示す．以降，図表中では類似プロジェクトの総工数の変動係数と分散についてはそれぞれ「総工数：変動係数」「総工数：分散」と，類似度の分散と平均は「類似度：分散」「類似度：平均」と表記する．

まず Euclidian Similarity を類似度として用いた場合について述べる．4種類の指標のうち，データセットに関わりなく常に有意に相関があったのは類似プロジェクトの総工数の分散である．どのデータセットの場合にも，有意水準 5% で相関がみられた．次に類似プロジェクトの総工数の変動係数が相関が強く，類似度の分散，類似度の平均と続いている．

表 4.1 ばらつきの各指標と MRE との相関係数（類似度：Euclidian Similarity）

		総工数： 変動係数	総工数： 分散	類似度： 分散	類似度： 平均
Desharnais データ	相関係数	0.137	0.279	-0.067	0.047
	有意差	0.236	0.014	0.056	0.685
Kitchneham データ	相関係数	0.278	0.224	0.385	0.400
	有意差	0.001	0.009	0.000	0.000
ISBSG データ	相関係数	0.308	0.433	0.105	0.100
	有意差	0.000	0.000	0.111	0.129

表 4.2 ばらつきの各指標と MRE との相関係数（類似度：Cosine Similarity）

		総工数： 変動係数	総工数： 分散	類似度： 分散	類似度： 平均
Desharnais データ	相関係数	0.210	0.298	-0.044	-0.093
	有意差	0.670	0.008	0.704	0.423
Kitchneham データ	相関係数	0.289	0.241	0.368	-0.392
	有意差	0.001	0.005	0.000	0.000
ISBSG データ	相関係数	0.325	0.424	0.091	-0.108
	有意差	0.000	0.000	0.168	0.101

この傾向は Cosine Similarity でも同様で，類似プロジェクトの総工数の分散がデータセットにかかわらず，常に有意に相関を示した．ただし，相関係数の値自体は，類似プロジェクトの総工数の分散では類似度の算出方法が Euclidian Similarity か Cosine Similarity かによる違いは見られなかった．

このことから，類似プロジェクトの総工数の分散と相対誤差 MRE には相関があり，相関係数が正の値を取っていることから，総工数の分散が大きいとき，MRE の値も大きくなり，見積りに失敗しやすくなることが分かった．ばらつきの指標と MRE の相関を踏まえ，次節では，MRE の値が 1 を超える場合に限定した，見積りの失敗とばらつきの指標との関係をグラフから検証する．

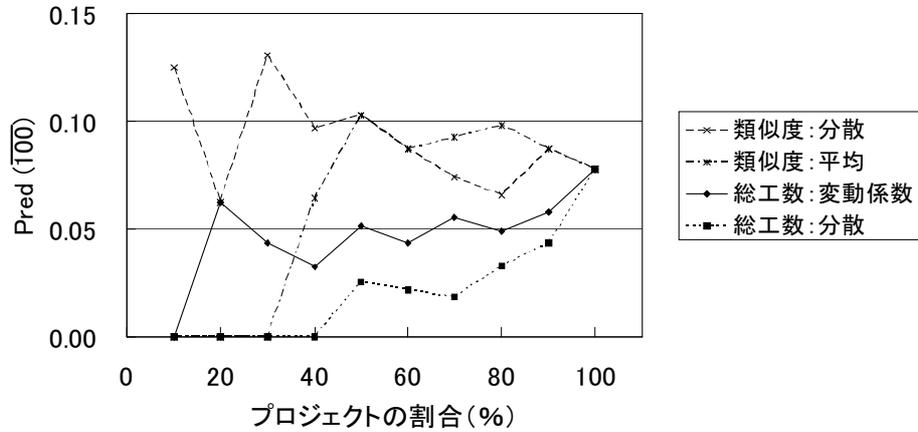


図 4.4 ばらつきの指標ごとの $\text{Pred}(\overline{100})$ の変化 (Desharnais データ, Euclidian Similarity)

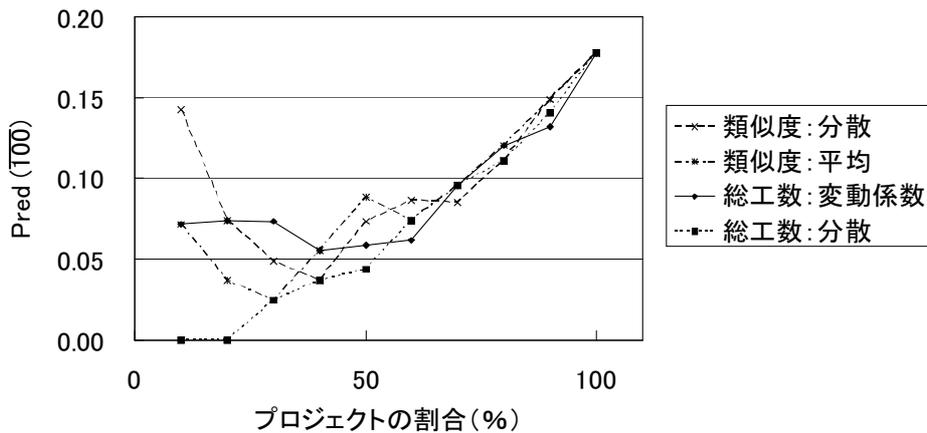


図 4.5 ばらつきの指標ごとの $\text{Pred}(\overline{100})$ の変化 (Kitchenham データ, Euclidian Similarity)

3.5.2 ばらつきと $\text{Pred}(\overline{100})$ の関係

分析から得られた $\text{Pred}(\overline{100})$ を図 4.4 ~ 図 4.9 に, MdMRE を図 4.10 ~ 図 4.15 に示す. 図中では横軸は指標の値の上位何パーセントが見積り結果として採用されたかを示す. ここでの上位とは, 各仮説において見積り精度が良くなると期待

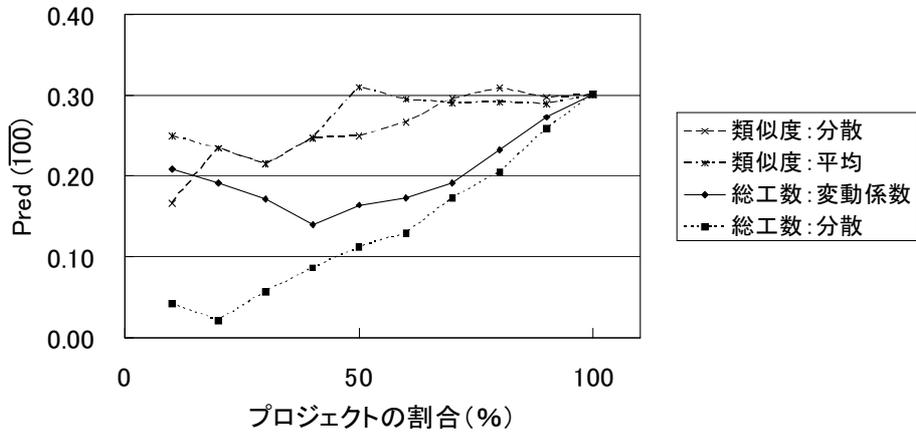


図 4.6 ばらつきの指標ごとの $\text{Pred}(\overline{100})$ の変化 (ISBSG データ , Euclidian Similarity)

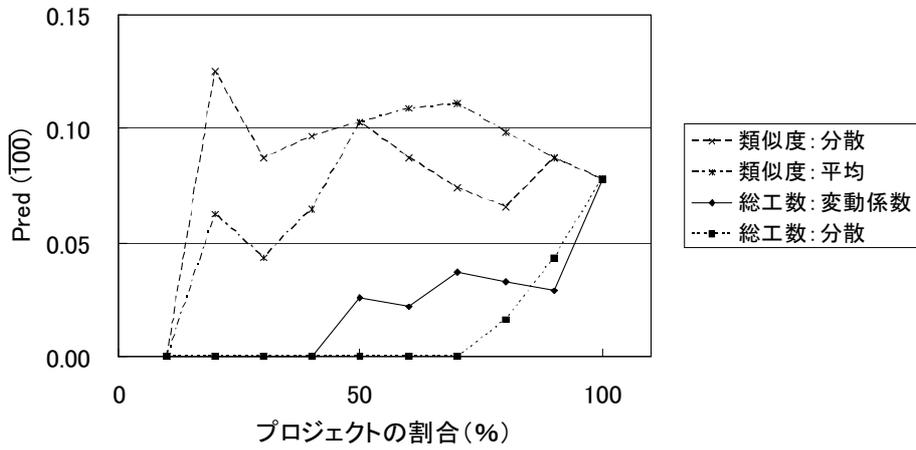


図 4.7 ばらつきの指標ごとの $\text{Pred}(\overline{100})$ の変化 (Desharnais データ , Cosine Similarity)

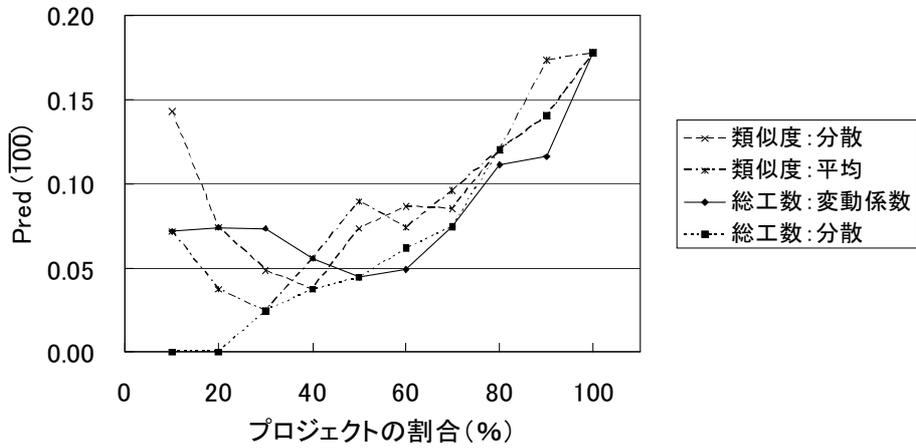


図 4.8 ばらつきの指標ごとの $\text{Pred}(\overline{100})$ の変化 (Kitchenham データ, Cosine Similarity)

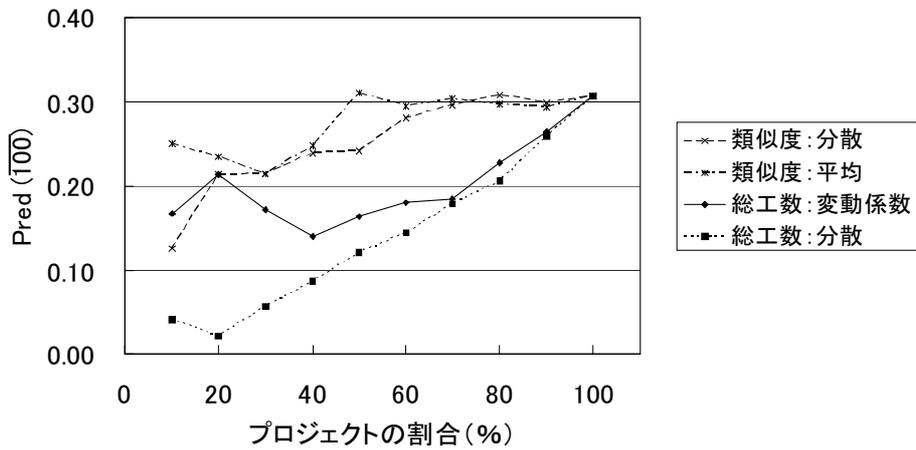


図 4.9 ばらつきの指標ごとの $\text{Pred}(\overline{100})$ の変化 (ISBSG データ, Cosine Similarity)

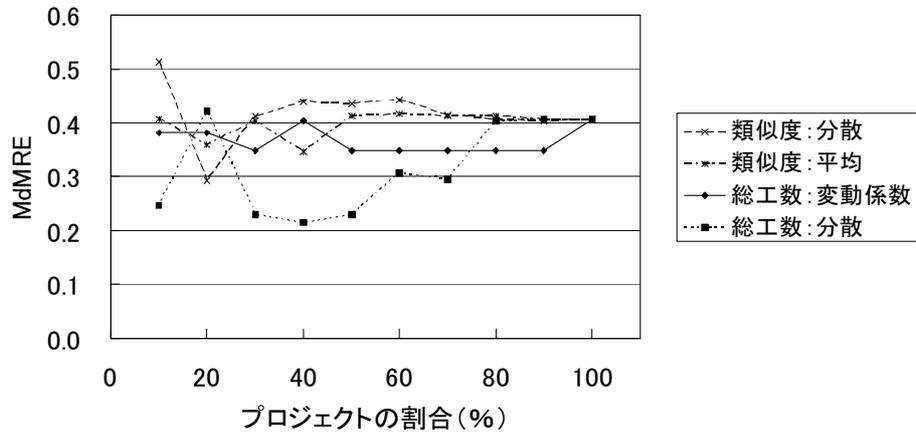


図 4.10 ばらつきの指標ごとの MdMRE の変化 (Desharnais データ, Euclidian Similarity)

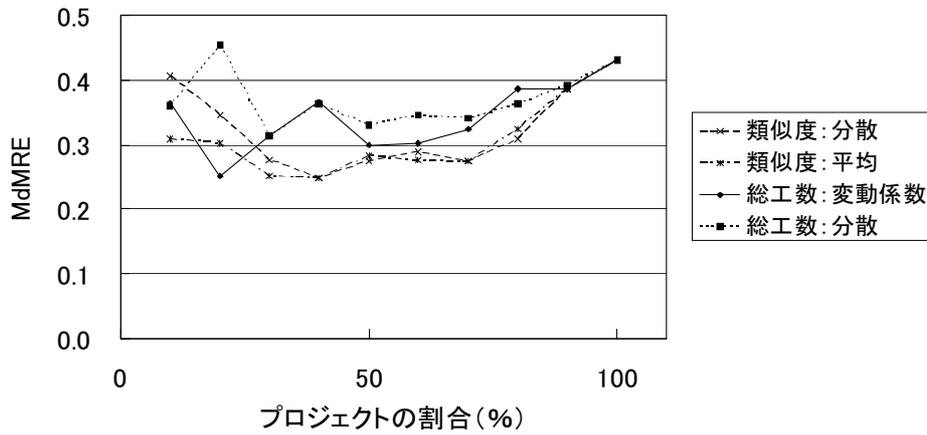


図 4.11 ばらつきの指標ごとの MdMRE の変化 (Kitchenham データ, Euclidian Similarity)

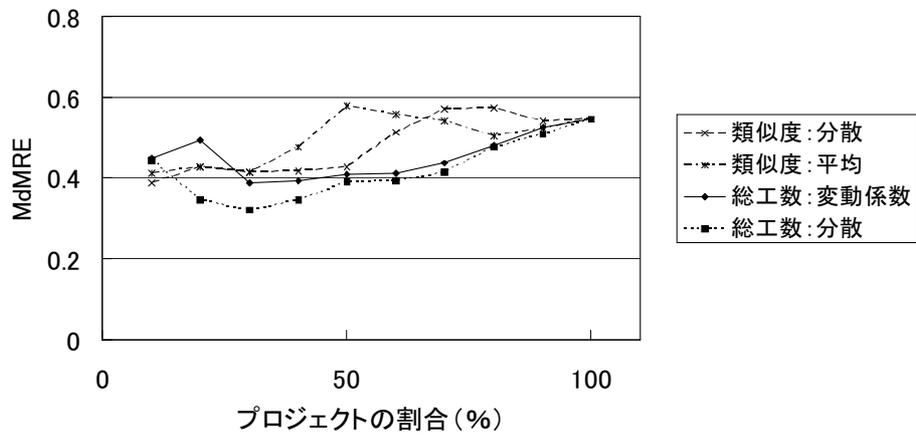


図 4.12 ばらつきの指標ごとの MdMRE の変化 (ISBSG データ , Euclidian Similarity)

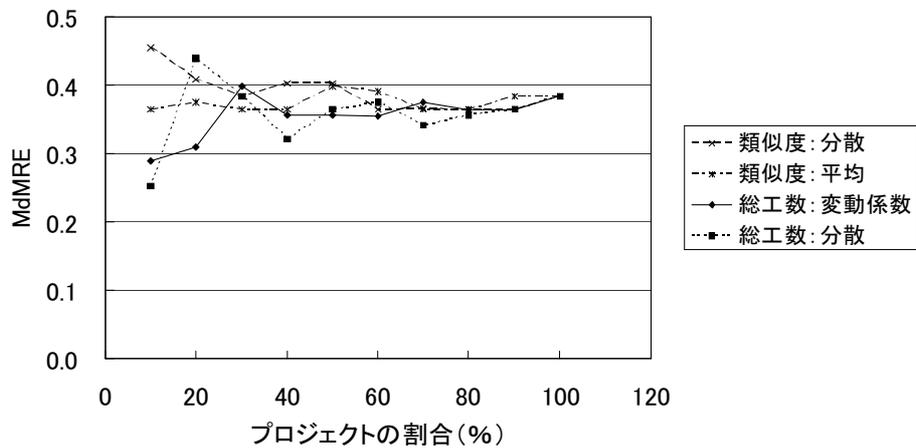


図 4.13 ばらつきの指標ごとの MdMRE の変化 (Desharnais データ , Cosine Similarity)

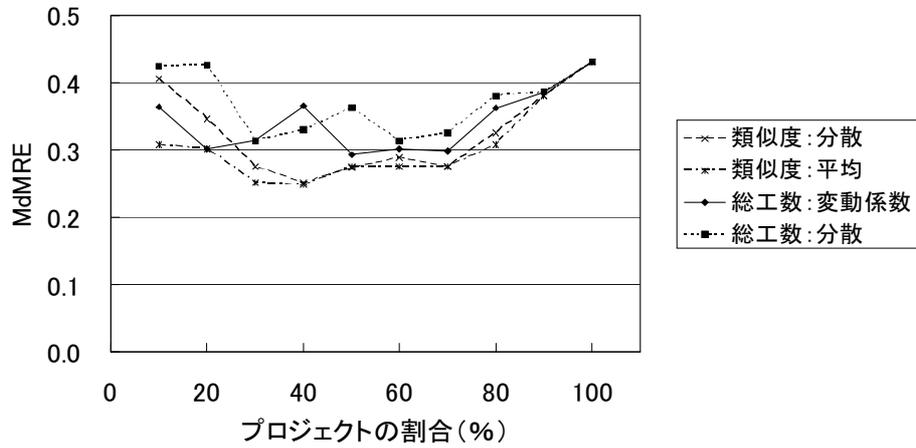


図 4.14 ばらつきの指標ごとの MdMRE の変化 (Kitchenham データ, Cosine Similarity)

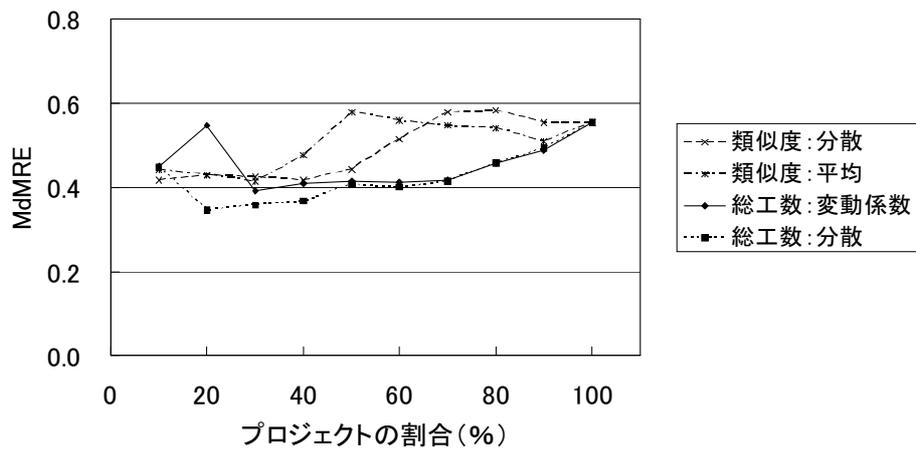


図 4.15 ばらつきの指標ごとの MdMRE の変化 (ISBSG データ, Cosine Similarity)

できるものを指し，類似プロジェクトの総工数の変動係数，分散では値が小さいものを，類似度のうち，Euclidian Similarity の分散，平均ではその値が小さいものを，Cosine Similarity ではその値が大きいものを上位とする．このため，横軸の値が小さくなるにつれて，指標の値は概して小さく（Cosine Similarity 関連に関しては大きく）なる．縦軸は $\text{Pred}(\overline{100})$ では相対誤差が 100%を超えるプロジェクトが採用された結果に占める割合を表し，MdMRE では採用された結果の MRE の中央値を表す．例えば $\text{Pred}(\overline{100})$ の図で，総工数の変動係数の縦軸の値が 0.05，横軸の値が 50 ならば，変動係数の値が上位 50%のプロジェクト中に相対誤差が 100%を超えているプロジェクトが占める割合が 0.05 すなわち 5%であることを示す．

まず $\text{Pred}(\overline{100})$ についての結果について述べる．全体として見た場合，MRE とばらつきの指標の相関と同様に， $\text{Pred}(\overline{100})$ は類似プロジェクトの総工数の分散と最も関係が深いことが分かった．データセットごとの結果では，図 4.4 から，Desharnais のデータでは類似度に Euclidian Similarity を用い，指標として総工数の分散を用いた場合，分散の値の上位（値が小さい）40%のプロジェクトは相対誤差が 100%を超えない．類似度の分散と平均がそれに続く．同様に図 4.5 から，Kitchenham データでは，総工数の分散を指標として採用すると，上位 20%のプロジェクトには相対誤差 100%以上のプロジェクトが含まれておらず，類似プロジェクトの総工数の分散は $\text{Pred}(\overline{100})$ との関係が深く，それ以外のものは，それほど深い関係ではないといえる．図 4.6 から，ISBSG データでは相対誤差が 100%を超えるプロジェクトが含まれなくなることはないものの，総工数の分散が最も $\text{Pred}(\overline{100})$ との関係が深く，総工数の変動係数が続いている．

類似度に Cosine Similarity を用いた場合，図 4.7 から，Desharnais のデータでは指標として総工数の分散を用いた場合，分散の値の上位（値が小さい）70%では相対誤差が 100%を超えるプロジェクトは存在しない．図 4.8 から，Kitchenham データでは，類似度が Euclidian Similarity だった場合とほぼ同一の結果となり，類似度の算出方法による違いはほとんどない．図 4.9 でも類似度の算出方法による違いはみられなかった．

これらの結果をまとめると， $\text{Pred}(\overline{100})$ との関係が最も深いのは類似プロジェ

クトの総工数の分散であり，類似度計算方法には Cosine Similarity を採用したほうが，Euclidian Similarity よりも関係はさらに深くなるが，その差はそれほど大きくはないことが分かった．

次に MdMRE について述べる．MdMRE とばらつきの各指標との間に明確な関係は見られなかった，データセットごとの結果では，類似度に Euclidian Similarity を用いた場合，Desharnais データセットでは，図 4.10 から上位 50%前後では総工数の分散に限り一時的に精度が向上するものの，それ以外の指標は精度に影響を与えていない．Kitchenham データセットでは，図 4.11 から一旦は指標の値が小さくなるにつれて精度が向上していくが，10%の場合のみ精度が悪くなっている類似度の平均など，精度の変化と各指標の間に相関はみられない．ISBSG データセットでは，図 4.12 からおおむねどの指標でも指標の値が小さくなるにつれて精度も向上している．しかしながら上位 30%以下では精度が悪くなっていく総工数の平均，上位 10%を結果に採用した場合のみ精度が急激に向上している Euclidian Similarity の分散など，どの指標が MdMRE と関係があるかを図から読み取るとは難しい．

類似度に Cosine Similarity を用いた場合にも，MRE と指標との間に明確な関係が見られないのは同様で，Desharnais データセットの結果である図 4.13 では，指標の値と MdMRE の間に関係は見られない．Kitchenham データの結果である図 4.14，ISBSG データの結果である図 4.15 でも，やはりほとんど関係は見られない．

3.6 考察

実験結果から，ばらつきを表す指標の一部については $\text{Pred}(\overline{100})$ すなわち見積りに失敗したプロジェクトが占める割合との間に関係があることが分かった．Desharnais データでは類似度に Euclidian Similarity を採用すれば，類似プロジェクトの総工数の分散が上位 40%のものには相対誤差が 100%を超えるプロジェクトが含まれていなかった．同様に Kitchenham データでは類似プロジェクトの総工数の分散が下位 20%に含まれているものでは，見積りに失敗したプロジェクトは存在しなかった．ISBSG データセットでは見積りに失敗したプロジェクトが存在

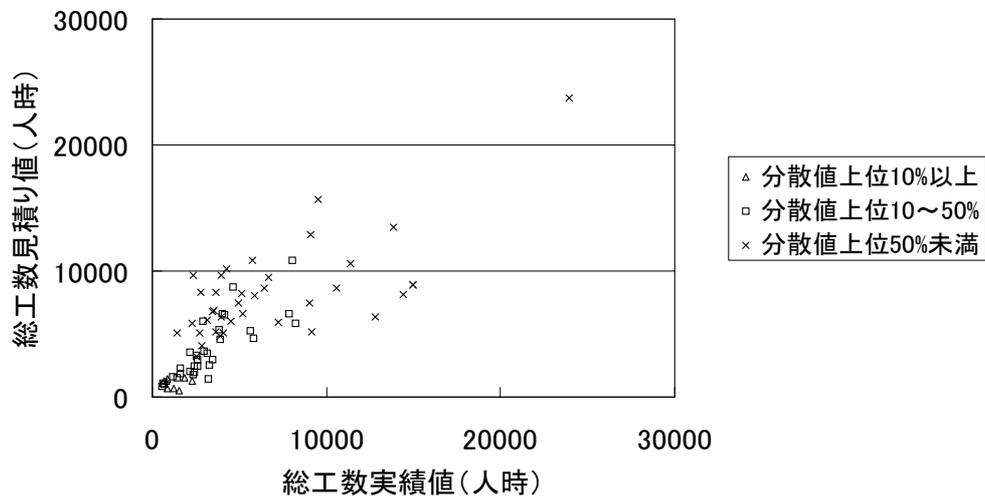


図 4.16 見積り対象の工数実績・見積り値と類似プロジェクトの工数分散値との関係 (類似度：Euclidian Similarity)

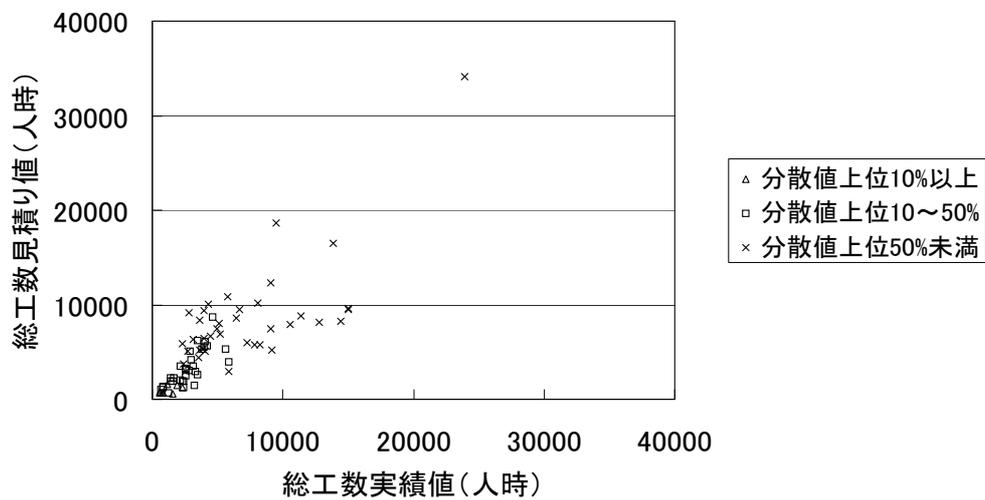


図 4.17 見積り対象の工数実績・見積り値と類似プロジェクトの工数分散値との関係 (類似度：Cosine Similarity)

しないような区分は得られなかったものの、類似プロジェクトの総工数と分散では常に、総工数の変動係数では50%程度までは、その値が小さくなるに従い見積りに失敗したプロジェクトが含まれる割合は単調減少していた。類似度に Cosine Similarity を採用すると、Desharnais データセットでは上位70%のものには相対誤差が100%を超えるプロジェクトが含まれていなかった。それ以外のデータセットでは Euclidian Similarity を採用した場合とほとんど同じであった。

類似度の算出方法に関わらず、全てのデータセットについて総工数の分散値と $\text{Pred}(100)$ の間に関係があることが示されたことを踏まえ、以下ではそのような関係が表れた理由について考察する。

実験では、類似度以外の指標として類似プロジェクト総工数の変動係数と分散を採用した。この2つの指標の違いとしては、分散が元の値が大きければこれらの算出値も大きくなる傾向にあるのに対し、変動係数は算出元の値に左右されにくい傾向がある、ということが挙げられる。類似度についても実験では標準化を行ったため変動係数と同様のことがいえる。このことから、そもそもアナロジーベース法では元の値、すなわち工数が大きいプロジェクトの見積りでは相対誤差が大きくなる傾向にあり、そのために類似プロジェクトの分散が大きい場合には見積り結果を採用しないほうが望ましいという結果が導かれた可能性がある。このような結果となった原因の1つとして、総工数の変動係数や分散の元となった値の特性が考えられる。総工数の変動係数や分散の算出に用いたのは、本章の場合は式3.3による補正後の値である。しかし、工数と規模との関係は直線的ではなく、規模の増大に対し、工数は指数関数的に増大する [4]。このためいくら規模で補正したとしても、見積り対象プロジェクトと類似プロジェクトの間の規模の差が大きい場合には見積りがうまくいかないことがあり得る。例として Desharnais データにおける見積り値と実績値と類似プロジェクトの分散値の関係を表すグラフを類似度に Euclidian Similarity を用いたものを図4.16に、Cosine Similarity を用いた者を図4.17に示す。横軸は実績値、縦軸は見積り値を表している。また図中「分散値上位10%」とは、その点が表すプロジェクトの類似プロジェクトの分散の値が、データセットに含まれる全プロジェクトの分散の値の上位10%以下に含まれているということを示す。図から明らかのように、平均、分散の値が小さ

くなるに従って工数の規模も小さくなっており，このことから，アナロジーベース法は，工数の大きい，すなわち規模の大きいプロジェクトにおいて見積りに失敗しやすいということが明らかとなった．そして大規模なプロジェクトは，小規模なプロジェクトに比べて見積りに失敗した場合の損害が大きくなるため，大規模なプロジェクトの見積りにはアナロジーベース法を用いることは避けるべきである．

4. まとめ

本章では，アナロジーベース法による工数見積りにおいて，「見積り対象プロジェクトと類似プロジェクト群の間の類似度の分散または平均が大きい場合，その見積りは失敗しやすい」「見積り対象に類似していると判定された類似プロジェクト群の工数の変動係数と分散が大きい場合，その見積りは失敗しやすい」という2つの仮説を立て，見積りに失敗しやすいプロジェクトを選定し，見積り対象から外す方法について検討した．そして選定結果に基づいて見積り回避を行った結果，見積りを大きく外すプロジェクトが減少することを実験から検証した．ケーススタディでは，類似プロジェクトの目的変数（総工数）の分散の値が大きい場合には見積り対象から外すべきであることが分かった．実験結果から，規模の大きいプロジェクトはアナロジーベース法での見積りには向かないことが分かった．このような結果になった原因の1つとして，規模が大きいプロジェクトは数が少ないということが考えられる．今後，規模の大きいプロジェクトを見積るのに適した方法と，アナロジーベース法を組み合わせることで見積り対象から外すべきプロジェクトを減らし，より外れる可能性の低い見積り方法を開発することが考えられる．また，提案方法では見積りが外れやすいプロジェクトは特定できるものの，アナロジー法では見積る意味が無い以上，特定したプロジェクトを別の方法で見積る必要があり，その方法は少なくともアナロジー法よりも見積りに失敗する可能性が低い方法でなくてはならない．このような条件を満たす見積り方法の選定が今後の課題となる．

第5章 アナロジーベース法と対数重回帰モデルを組み合わせた工数見積り方法の提案

1. まえがき

4章から、アナロジーベース法において、見積り対象のプロジェクトに類似していると判定されたプロジェクト群の総工数の分散が大きい場合、見積りは失敗しやすいことが分かった。しかしながらこの方法を実際の開発現場で用いた場合、1つの問題が発生する。見積りに失敗する可能性が高いプロジェクトを判別できたとしても、それはその見積り値が失敗する値から空白に変化するだけであり、見積りが行われなかった場合と変わらない。したがって、4章で提案した方法を用いた結果、見積りを回避すべき（見積りに失敗する）と判定されたプロジェクトは、別の工数見積り方法を採用し、見積り値を得る必要がある。そして、その見積り値は少なくとも見積りに失敗すると判定された見積り値よりも、見積りに失敗する可能性が低い、または精度が高いことが期待できなくてはならない。そこで本章では、見積りに失敗すると判定されたプロジェクトを、アナロジーベース法とは別の方法、具体的には対数重回帰モデルで見積ることで、見積りに成功するプロジェクトはそのままに、見積りに失敗するプロジェクトではアナロジーベース法よりも見積りに失敗しにくい見積り値を得るというハイブリッドな見積り方法を提案する。また本章では4章の結果を踏まえ、見積りに失敗するプロジェクトの判別には類似プロジェクトの総工数の分散を用いた。類似度については4章と同様に、Euclidian Similarity と Cosine Similarity の両方を採用し、それぞれ実験に用いた。

以降5章2節で提案方法の性能検証のためのケーススタディを行い，提案方法の概要とその手順，そしてケーススタディの結果とその考察を行う．5章3節ではそれらのまとめについて述べる．

2. ケーススタディ

2.1 概要

4章3節の実験結果から，目的変数（総工数）分散は $\text{Pred}(\overline{100})$ との間に深い関係があり，分散の値が大きい場合，見積りに失敗しやすいことが分かった．しかしながら，4章3節の実験では，ばらつきの指標，本章では総工数の分散の値が大きいプロジェクトを実験結果から除外してゆき，値が小さいプロジェクトのみが残った場合についての議論は行ったが，除外されたプロジェクトに対してどのように対処すべきかについては一切触れてこなかった．そこで，除外されたプロジェクトに対するケアとして，対数重回帰モデルを構築，見積りを行い，除外されなかったプロジェクトではアナロジーベース法で見積りを行うというハイブリッドな方法（以降複合法と記述する）を考える．ただし，類似プロジェクトの総工数の分散が大きいプロジェクトについても見積りを行う以上，先ほどの実験結果のような $\text{Pred}(\overline{100})$ が0となるような状況になることはほとんどない．そこで，ここではアナロジーベース法を比較対象とし，アナロジーベース法に対する複合法での $\text{Pred}(\overline{100})$ の減少（失敗プロジェクトの低減）について着目する．

2.2 手順

4章3節での実験結果に対数重回帰モデルによる見積り結果を追加する形で見積りを行った．ただし，4章との相違点として，相関分析についての部分は省略されている．実験の手順を以下に述べる．

手順1 データセットに対してリーブワンアウトで対数重回帰モデルを構築し，各プロジェクトについての見積り値を得る．

手順 2 類似プロジェクトの総工数の分散について，対数重回帰モデルによる見積りを行うか，アナロジーベース法による見積りを行うかの閾値を決定する．ここでは初期値を 10% とする．

手順 3 プロジェクトの 1 つを選ぶ．4 章の実験結果で，類似プロジェクトの総工数の分散の値が全体の上位 10% に含まれているならばアナロジーベース法による見積り結果を，そうでないならば対数重回帰モデルによる見積り結果を採用する．

手順 4 手順 3 を全てのプロジェクトについて行う．

手順 5 手順 4 の結果を，アナロジーベース法の割合が 10% の場合の複合法の見積り結果とする．

手順 6 上記，手順 1 から手順 5 を，閾値を 10% ずつ増やし，アナロジーベース法の割合が 100% になるまで行う．

手順 7 全プロジェクトの結果から得られた MRE を用いて，アナロジーベース法が占める割合ごとの $\text{Pred}(\overline{100})$ および MmMRE を算出する．

2.3 結果

実験から得られた $\text{Pred}(\overline{100})$ を図 5.1 ~ 図 5.6 に，MmMRE を図 5.7 ~ 図 5.12 に示す．図中の縦軸は先の実験と同様である．横軸は類似プロジェクトの総工数の分散値の上位（値が小さい）何パーセントがアナロジーベース法による見積り結果を採用したかを示し，残りは対数重回帰モデルによる見積り結果が採用されていることを示す．横軸が 30% の場合，類似プロジェクトの分散の値が上位 30% のものについてはアナロジーベース法による見積り結果が採用され，残りの 70% は対数重回帰モデルによる見積り結果が採用されている．図中ではアナロジーベース法のみでの見積り結果を「アナロジー法」と記述している．

類似度算出方法，データセットに関わらず，提案する複合法はアナロジーベース法での見積りに比べて $\text{Pred}(\overline{100})$ ，MmMRE とともに顕著な減少を示した．デー

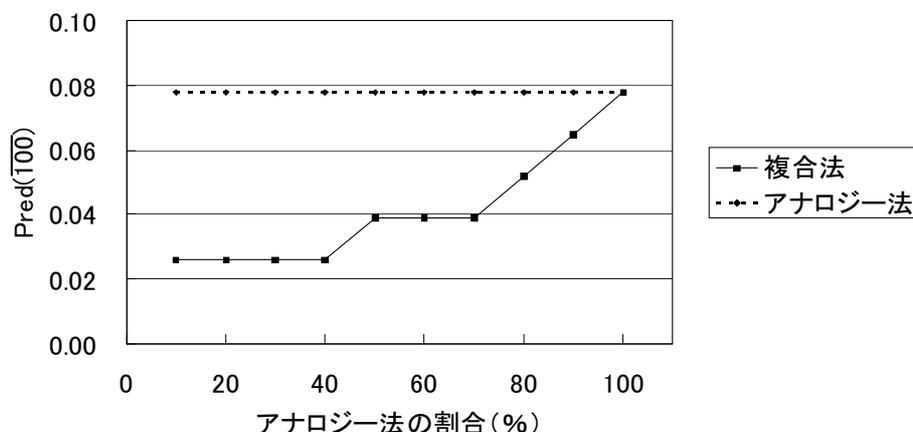


図 5.1 アナロジーベース法と複合法の $\text{Pred}(\overline{100})$ の比較 (Desharnais データ, Euclidian Similarity)

タセットごとに見た場合, Desharnais データセットでは類似度が Euclidian Similarity を用いた場合, 上位 70%まで単調減少し, 次に上位 40%でふたたび減少する以外は減少傾向を示さなかった. 類似度に Cosine Similarity を用いた場合には 70%まで単調減少し, それ以降は減少しなかった. MdMRE では上位 20% ~ 70%間では, Euclidian Similarity の場合の法が精度が高くなる傾向が見られた. Kitchenham データセット, ISBSG データセットでは類似度算出方法による $\text{Pred}(\overline{100})$, MdMRE の増減に変化はほとんど見られなかったが, アナロジーベース法のみを用いた見積りに比べて $\text{Pred}(\overline{100})$ の低減, MdMRE の減少 (全体的な精度の向上) が確認できた.

2.4 考察

ケーススタディの結果から複合法の実用性についての考察を行う. まず $\text{Pred}(\overline{100})$ に着目する. 類似度として Euclidian Similarity を用いた場合, Desharnais データセットでは上位 70%までと, 上位 40 ~ 50%間以外では減少傾向を示さなかった. これは上位 70%までの減少で $\text{Pred}(\overline{100})$ の値があまりに小さくなった (上位 70%までで 4%以下, 上位 40%以下で 3%以下) ことと, 元のデータセットのサ

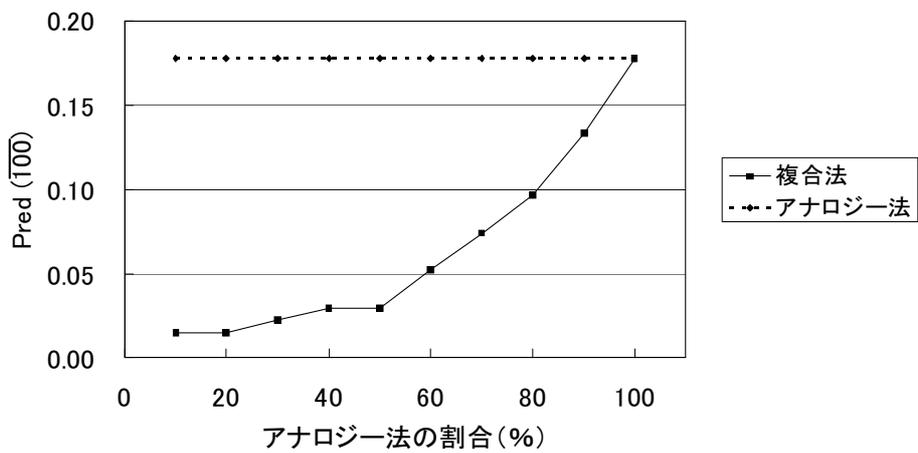


図 5.2 アナロジーベース法と複合法の $\text{Pred}(\overline{100})$ の比較 (Kitchenham データ, Euclidian Similarity)

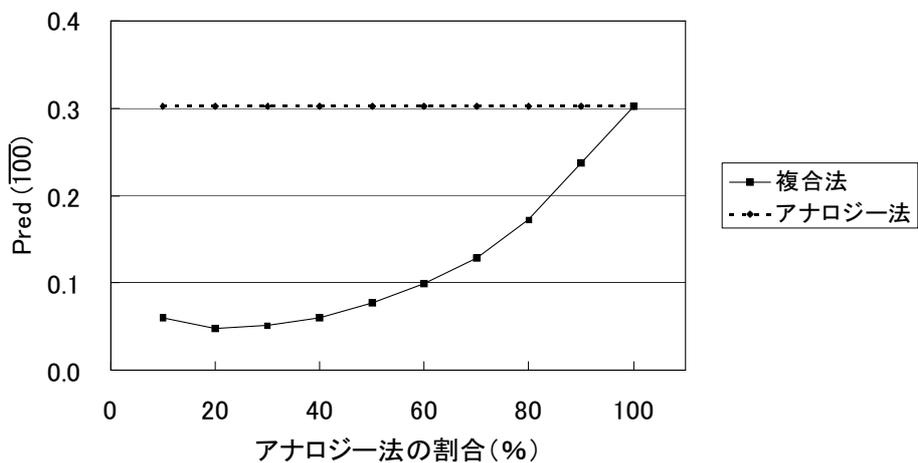


図 5.3 アナロジーベース法と複合法の $\text{Pred}(\overline{100})$ の比較 (ISBSG データ, Euclidian Similarity)

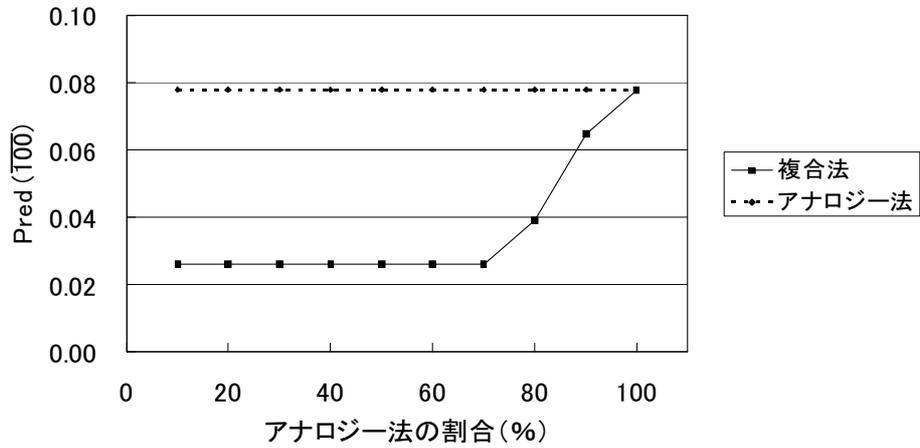


図 5.4 アナロジーベース法と複合法の $\text{Pred}(\overline{100})$ の比較 (Desharnais データ, Cosine Similarity)

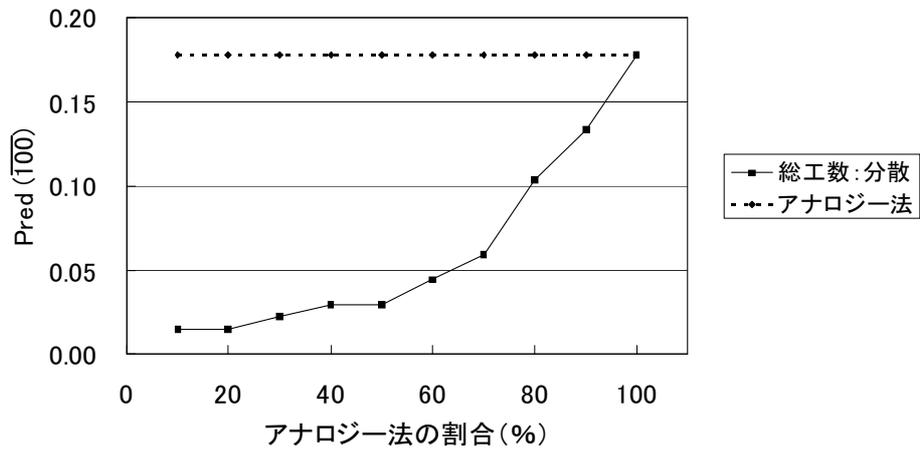


図 5.5 アナロジーベース法と複合法の $\text{Pred}(\overline{100})$ の比較 (Kitchenham データ, Cosine Similarity)

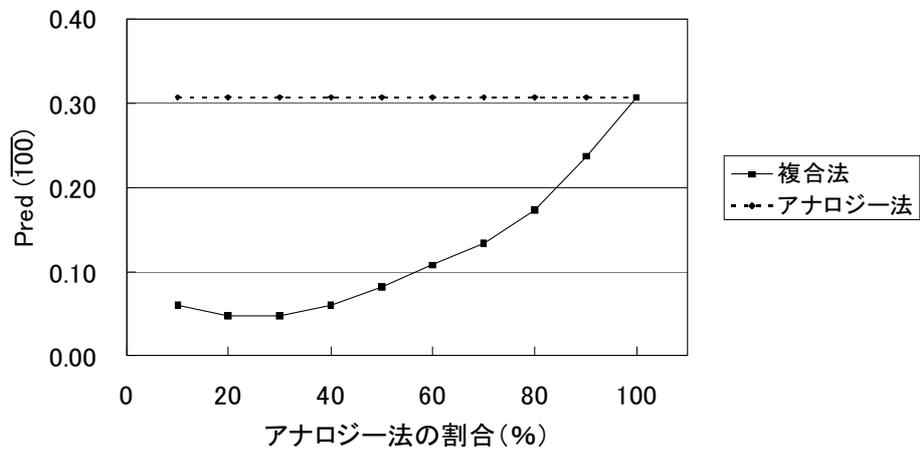


図 5.6 アナロジーベース法と複合法の $\text{Pred}(\overline{100})$ の比較 (ISBSG データ , Cosine Similarity)

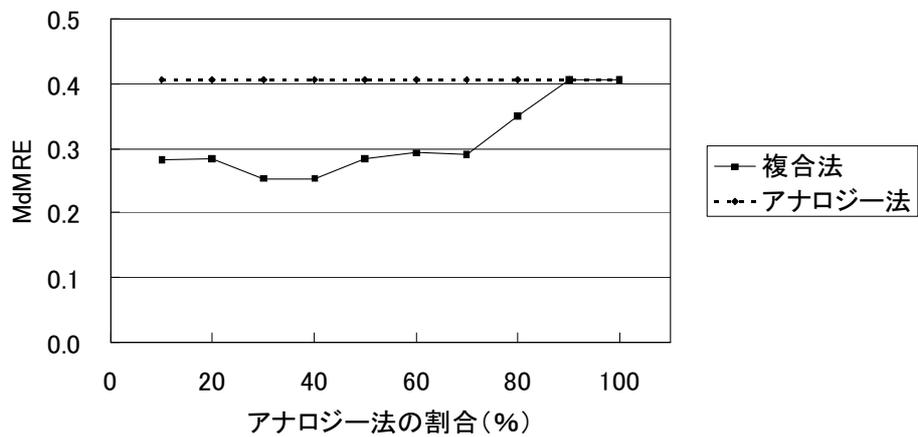


図 5.7 アナロジーベース法と複合法の MdMRE の比較 (Desharnais データ , Euclidean Similarity)

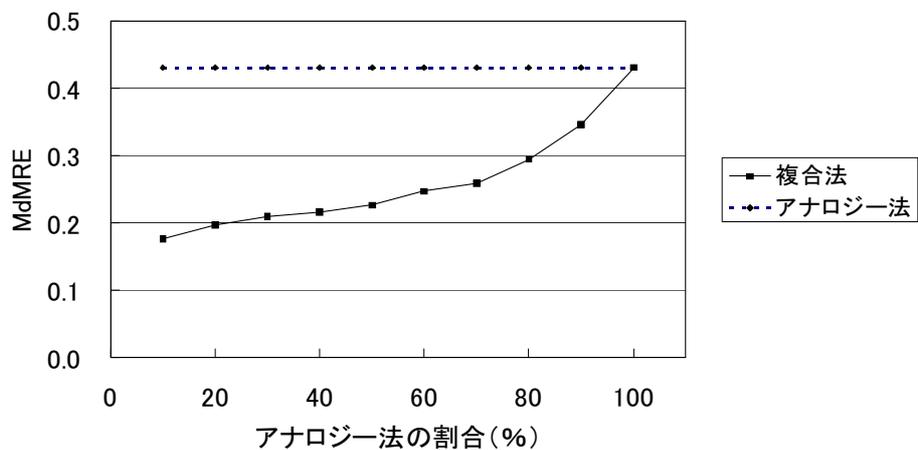


図 5.8 アナロジーベース法と複合法の MdMRE の比較 (Kitchenham データ, Euclidian Similarity)

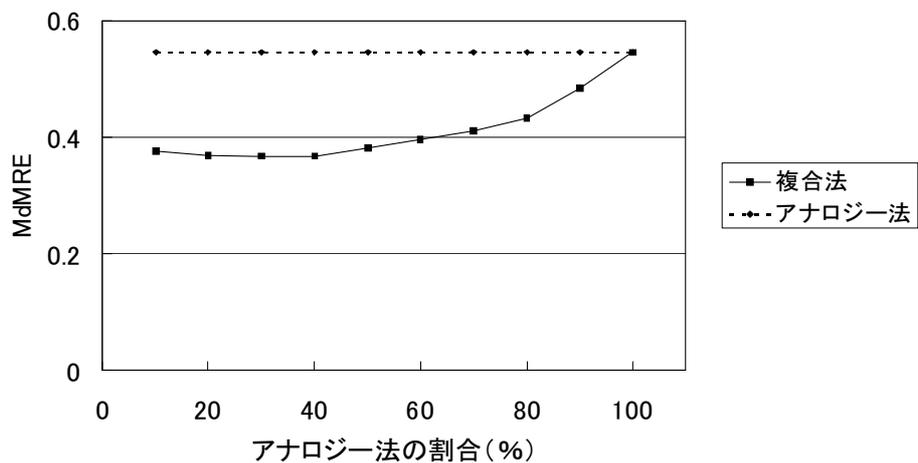


図 5.9 アナロジーベース法と複合法の MdMRE の比較 (ISBSG データ, Euclidian Similarity)

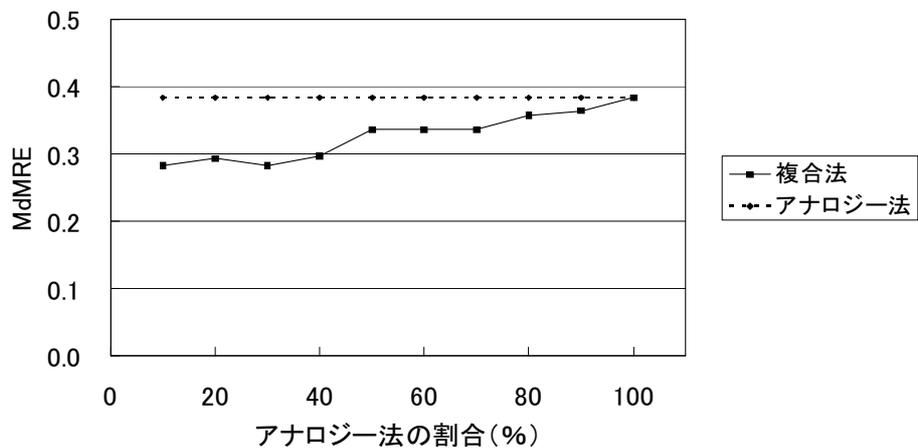


図 5.10 アナロジーベース法と複合法の MdMRE の比較 (Desharnais データ, Cosine Similarity)

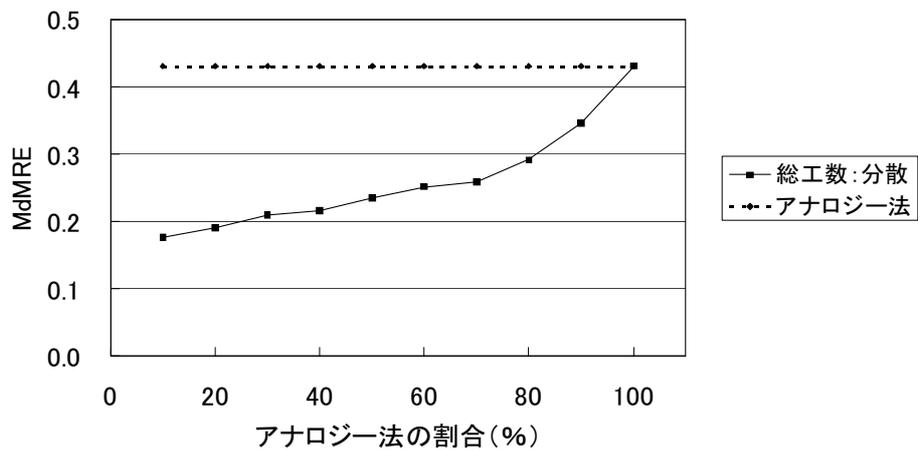


図 5.11 アナロジーベース法と複合法の MdMRE の比較 (Kitchenham データ, Cosine Similarity)

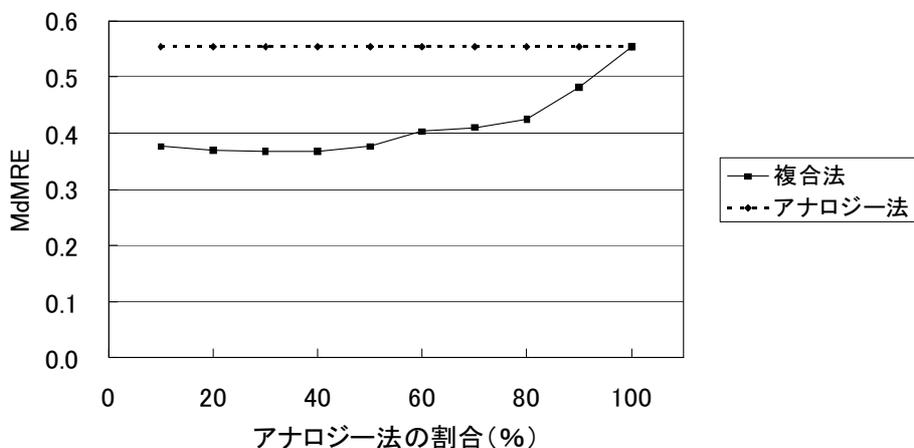


図 5.12 アナロジーベース法と複合法の MdMRE の比較 (ISBSG データ , Cosine Similarity)

イズが 77 件であり他の 2 つと比べて小さいためにそれ以上の減少が難しくなった可能性がある。類似度として Cosine Similarity を採用した場合にも同様の傾向が見られた。Cosine Similarity では上位 70%の時点で $Pred(\overline{100})$ は 3%以下となり、それ以上の減少が難しくなったと考えられる。次に MdMRE に着目する。同様に Desharnais データに着目すると、類似度としては Cosine Similarity よりも Euclidian Similarity を用いた方が MdMRE の値は小さく (精度は高く) なりやすい。しかしながら $Pred(\overline{100})$ の結果から、Cosine Similarity を用いた方が見積りに失敗しにくいいため、どちらを採用すべきかは、利用者がその状況に応じて決めるべきだと考えられる。それ以外のデータセットでは、複合法はアナロジーベース法に対して類似度の算出方法、精度評価指標に関わらず、顕著な精度の向上が見られた。これらの結果から、複合法はアナロジーベース法よりも優れた方法であると言える。

3. まとめ

本章ではアナロジーベース法では見積りが失敗しやすいプロジェクトに対して対数重回帰モデルによる見積りを行い、失敗しにくいプロジェクトについてはそ

のままアナロジーベース法を適用する複合的な見積り方法を提案した。提案方法によって、アナロジーベース法で見積りを行うよりも見積りに失敗する確率を低減することが可能であることが示され、同時に高い精度で見積り値を得ることができることも明らかとなった。ケーススタディでは、アナロジーベース法でまず類似プロジェクトを検出し、類似していると判定された複数のプロジェクトについて、目的工数（総工数）の分散値が大きい場合、対数重回帰モデルを、そうでない場合にはそのままアナロジーベース法を用いて見積りを行った。実験の結果、アナロジーベース法のみで見積りを行う場合に比べて提案方法は高い精度で見積りことができ、同時に見積りに失敗する可能性の低いことが示されたと考えられる。今後、アナロジーベース法と組み合わせる見積り方法を変更することで、さらなる精度の改善が期待できる。

第6章 おわりに

本論文では実用に耐える定量的工数見積り方法という目的のために、平均的な見積り精度の向上と見積りを大きく外すプロジェクトの低減という2つのアプローチを採った。定量的工数見積り方法のうち、従来広く用いられてきた工数見積り方法である線形重回帰モデルとアナロジーベース法に対し、開発プロジェクトの個別性を考慮した拡張をそれぞれ行い、アプローチの実現を目指した。まず、工数見積りの見積り精度のばらつきとプロジェクト特性の関係を明らかにするための分析を行った。その結果、開発言語や開発工数の計測方法が精度との関わりが深いことが分かった。さらに、精度との関わりが深い特性に注目し、そのプロジェクト特性を含むプロジェクトだけを集めて(層別して)重回帰モデルを構築、見積りをする場合と、それ以外のプロジェクト特性も含めてモデルを構築、見積りをする場合とで比較した。その結果、層別によって精度の向上が見られた特性もあるものの、層別を行わない方が精度が高い特性も存在した。この結果は層別が常に精度の向上をもたらすとは限らない可能性があることを示唆している。また層別を行うにしても、プロジェクトは通常、複数の特性を持つため、どの特性を層別に用いることが高精度の見積り結果をもたらすかは分からない。そこで次の研究として、過去のプロジェクトを見積り対象プロジェクトが含む特性ごとに層別し、そのうちで最も見積り精度が高くなることが想定されるフィットデータを選定する方法を提案、検証した。ただし先の結果を踏まえ、選定対象には層別前のフィットデータも加えた。その結果、提案方法は層別を行わずに見積りをした場合に比べて高い精度での見積りが可能であることが示された。また選定結果から、いくつかのプロジェクトにおいては全てのプロジェクト(層別前のフィットデータ)が採用されていることが分かった。このことから、プロジェクトによっては層別を行わないでモデルを構築、見積りを行ったほうが精度が高くなる場合

があることが明らかとなった。

最後に、見積りに失敗するプロジェクトを回避するための研究を行った。アナロジーベース方法において類似度の平均や分散、類似プロジェクトの工数の変動係数や分散が大きければ見積りに失敗する可能性が高いという仮説を立て、検証した。その結果、類似プロジェクトの工数の分散が大きい場合には見積り自体を回避すべきであることが分かった。さらに見積りを回避すべきと判定されたプロジェクトについては対数重回帰モデル方法を用い、見積り回避プロジェクトではないものについてはそのままアナロジーベース方法を用いるという複合的な方法を提案、検証した。その結果、アナロジーベース方法に比べて概して見積りに失敗する確率は下がり、全体的な見積り精度も高くなることが示された。

本論文で提案した2つの方法を、見積りを行う者の要求に合わせて用いることにより、状況に合わせた見積りが可能となり実用に耐える工数見積りが可能となる。

謝辞

これまで行ってきた研究が博士論文という形となるまでに、多くの方からご指導、ご協力を頂きました。本論文に直接、間接を問わず、関わっていただいた全ての方に感謝の意を述べさせていただきます。ありがとうございました。

以降の謝辞では、その中でも特に多くの協力を頂いた方へ改めて感謝の意を捧げるものです。

奈良先端科学技術大学院大学 情報科学研究科 松本 健一 教授には、研究への直接の指導のみならず、研究へ向き合う姿勢、原稿の書き方等、さまざまな面で熱心な御指導を賜りました。また、先生が整えられた、非常に充実した研究環境は、研究活動に専念する上での大きな助けとなりました。先生の厳しくも温かいご指導と、すばらしい研究環境無しにはこの研究は成し得なかったと確信しております。本論文に限定しても、研究課題や方針について数多くのご指摘、ご指導を頂きました。心より厚く御礼申し上げます。

奈良先端科学技術大学院大学 情報科学研究科 関 浩之 教授には、本研究について、その研究課題や方針についての的確な御助言、御指摘を数多く頂きました。その多くは私に欠けていた視点からのものであり、先生の御指導により、論文の充実と、洗練が為されました。心より厚く御礼申し上げます。

奈良先端科学技術大学院大学 情報科学研究科 飯田 元 教授には、本研究に際してさまざまなご助言を賜るとともに、研究活動においてもさまざまなご指導を賜りました。先生のご指導により、研究活動を円滑に進めることができました。また、先生のお人柄により、私の学生生活はより明るいものとなりました。心より感謝いたします。

奈良先端科学技術大学院大学 情報科学研究科 門田 暁人 准教授には、研究における議論や、問題解決のためのアイデアの提示に始まり、研究の進め方、論文執筆に至るまで、研究生生活の全てに渡り、熱意ある御指導を賜りました。特に、原稿投稿や研究発表の直前において、昼夜を問わず非常に丁寧、かつ厚い御指導を賜りました。また、先生のお人柄により、私の学生生活はより明るいものとなりました。私の研究は、先生の御指導は成しえませんでした。心より厚く御礼申し上げます。

奈良先端科学技術大学院大学 情報科学研究科 大平 雅雄 助教には，研究者としての在り方について多くの御助言を賜りました．研究の楽しさ，厳しさ，研究に対する姿勢など数多くのことを学ぶことができました．研究以外でも大変親しく接していただき，私の学生生活はより充実したものとなりました．心より厚く御礼申し上げます．

奈良先端科学技術大学院大学 情報科学研究科 森崎 修司 助教には，先生の研究に対する姿勢から，数多くのことを学ばせていただきました．また研究以外でも，効率を追求する徹底した時間管理やプロジェクト管理などについて学ばせていただきました．心より厚く御礼申し上げます．

奈良先端科学技術大学院大学 情報科学研究科 川口 真司 助教には，研究活動や学生生活に関する貴重な御助言を賜りました．先生のご助言は研究活動に大いに助けとなりました．また研究以外でも大変親しく接していただき，私の学生生活は充実したものとなりました．心より厚く御礼申し上げます．

奈良先端科学技術大学院大学 情報科学研究科 角田 雅照 特任助教には，本研究の核となる事柄について多くの御指摘，御助言を賜りました．また，統計学全般についての私の疑問に，貴重なお時間を割き，明確かつ丁寧に回答を頂きました．研究以外でも大変親しく接していただき，私の学生生活は充実したものとなりました．心より厚く感謝いたします．

神戸大学大学院 工学研究科 中村 匡秀 准教授には，研究論文の書き方や研究会などでの発表の作法など，研究者に必要な素養の習得について御指導を賜りました．また，学生生活において数多くの御助言を賜りました．研究以外でも大変親しく接していただきました．心より厚く御礼申し上げます．

神戸大学大学院 工学研究科 井垣 宏 特命助教には，研究との関係の有無を問わず，様々な相談に乗っていただき，多くのアドバイスをいただきました．心より厚く御礼申し上げます．

大阪大学大学院 情報科学研究科 柿元 健 特任助教には，本研究を進めるにあたり，研究の課題についてご助言を頂きました．また，研究発表や論文執筆についても多くの丁寧な御指導を賜りました．心より厚く御礼申し上げます．

奈良工業高等専門学校 情報工学科 上野 秀剛 助教には幾度も研究に関する御

助言を賜りました。大学院入学以来、多くの時間を共有していただき、様々な事を学ばせていただきました。深く感謝いたします。

奈良先端科学技術大学院大学 ソフトウェア工学講座 亀井 靖高 博士、松本 真佑 氏には、研究に関する多くの御助言を賜りました。研究生生活の多くを共有していただき励ましてくださいました。深く感謝いたします。

奈良先端科学技術大学院大学 ソフトウェア工学講座 乾 純子 氏には、研究の遂行に必要な事務処理など多岐に渡り手助けいただきました。深く感謝いたします。

奈良先端科学技術大学院大学 ソフトウェア工学講座 山内 寛己 氏には、本研究を進めるにあたり、研究環境の整備の面で御助力を賜りました。また研究以外でも、様々な面での御助力を頂きました。深く感謝いたします。

奈良先端科学技術大学院大学 ソフトウェア工学講座ならびにソフトウェア設計学講座の皆様には、研究や日々の生活で様々な御助力を賜りました。非常に充実した研究生生活を過ごすことができたのは、ひとえに皆様のお陰です。特に、大蔵 君治 氏（現 同大学院 博士後期課程）、伏田 享平 氏（現 同大学院 博士後期課程）には、平日頃より楽しい時間を共有させていただきました。心より厚く御礼申し上げます。

最後に、日頃より私を支えてくれた家族に心より感謝します。

参考文献

- [1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, Vol. 19, No. 6, pp. 716–723, 1974.
- [2] A. Albrecht and J. Gaffney. Software function, source lines of code, and development effort prediction. *IEEE Transactions on Software Engineering*, Vol. 9, No. 6, pp. 83–92, 1979.
- [3] L. Angelis and I. Stamelos. A simulation tool for efficient analogy based cost estimation. *Empirical Software Engineering*, Vol. 5, No. 1, pp. 35–68, 2000.
- [4] B. Boehm. *Software Engineering Economics*. Prentice Hall, 1981.
- [5] L. Briand and W. Jürgen. Modeling development effort in object-oriented systems using design properties. *IEEE Transactions on Software Engineering*, Vol. 27, No. 11, pp. 963–986, 2001.
- [6] L. Briand, T. Langley and I. Wiecek. A replicated assessment and comparison of common software cost modeling techniques. Proceedings of 22th International Conference on Software Engineering (ICSE '00), pp. 377–386, 2000.
- [7] S. Conte, H. Dunsmore and Y. Shen. *Software Engineering Metrics and Models*. The Benjamin-Cummings Pub. Co., 1986.
- [8] J. Desharnais. Analyse statique de la productivité des projets informatiques à partir de la technique des points de fonction. Master's thesis, University of Montreal, 1989.
- [9] R. Gray, S. G. Macdonell and M. J. Sheppard. Factors systematically associated with errors in subjective estimates of software development effort: The stability of expert judgement. In *Proceedings of 6th International Software Metrics Symposium (METRICS'99)*, pp. 216–227, Boca Raton, November 1999.

- [10] International Software Benchmarking Standards Group. *The Benchmark Release 6*. ISBSG, 2000.
- [11] International Software Benchmarking Standards Group. *Estimating Benchmarking and Research Suite Release 9*. ISBSG, 2004.
- [12] D. Hawkins. The problem of overfitting. *Journal of Chemical Information and Modeling*, Vol. 44, No. 1, pp. 1–12, 2004.
- [13] Project Management Institute. *A Guide to the Project Management Body of Knowledge (PMBOK Guide) 2000 Edition*. Project Management Institute, 2000 ed edition, 12 2000.
- [14] ISBSG. International software benchmarking standards group. <http://www.isbsg.org/>.
- [15] R. Jeffery, M. Ruhe and I. Wiczorek. Using public domain metrics to estimate software development effort. *Proceeding of 7th International Software Metrics Symposium (METRICS '01)*, pp. 16–27, 2001.
- [16] G. Kadoda, M. Cartwright, L. Chen and M. Shepperd. Experience using case-based reasoning to predict software project effort. In *Proceeding of 4th International Conference on Empirical Assessment & Evaluation in Software Engineering*, pp. 1–34, 2000.
- [17] B. Kitchenham, G. MacDonell, M. Pickard and M. Shepperd. What accuracy statistics really measure. *IEE Proceeding Software*, Vol. 14, No. 3, pp. 1–85, 2001.
- [18] B. Kitchenham, S. Pfleeger, B. McColl and S. Eagan. An empirical study of maintenance and development estimation accuracy. *Journal of Systems and Software*, Vol. 64, No. 1, pp. 57–77, 2002.

- [19] J. Kromrey and C. Hines. Nonrandomly missing data in multiple regression: An empirical comparison of common missing-data treatments. *Educational and Psychological Measurement*, Vol. 54, No. 3, pp. 573–593, 1994.
- [20] C. Lokan. What should you optimize when building an estimation model? Proceeding of 11th IEEE International Software Metrics Symposium (METRICS '05), p. 34, 2005.
- [21] C. Mair, G. Kadoda, M. Lefley, K. Phalp, C. Schofield, M. Shepperd and S. Webster. An investigation of machine learning based prediction systems. *Journal of Systems and Software*, Vol. 53, No. 1, pp. 23–29, 2000.
- [22] Steve McConnell. *Software Estimation: Demystifying the Black Art (Best Practices (Microsoft))*. Microsoft Press, 3 2006.
- [23] E. Mendes and B. Kitchenham. Further comparison of cross-company and within-company effort estimation models for web applications. Proceeding of 11th IEEE international Software Metrics Symposium (METRICS '05), pp. 348–357, 2004.
- [24] E. Mendes, C. Lokan, R. Harrison and C. Triggs. A replicated comparison of cross-company and within-company effort estimation models using the isbsg database. Proceeding of 11th IEEE International Software Metrics Symposium (METRICS '05), p. 36, 2005.
- [25] E. Mendes, I. Watson, C. Triggs, N. Mosley and S. Counsell. A comparative study of cost estimation models for web hypermedia applications. *Empirical Software Engineering*, Vol. 8, No. 2, pp. 163–196, 2003.
- [26] O. Mizuno, T. Kikuno, K. Inagaki, Y. Takagi and K. Sakamoto. Statistical analysis of deviation of actual cost from estimated cost using actual project data. *Information and Software Technology*, Vol. 42, No. 7, pp. 445–515, 2000.

- [27] S.Murray (著), 氏家勝巳, 土井誠 (訳). マグロウヒル大学演習 統計, p. 73. オーム社, 1995.
- [28] Naoki Ohsugi, Masateru Tsunoda, Akito Monden and Ken ichi Matsumoto. Effort estimation based on collaborative filtering. In F. Bomarius and H. Iida, editors, *Proceeding of 5th International Conference on Product Focused Software Process Improvement (Profes2004)*, Vol. 3009, pp. 274–286, 2004.
- [29] L. Putnum and W. Myers. *Measures for Excellence*. PRENTICE HALL, 1992.
- [30] M. Jørgensen. Experience with the accuracy of software maintenance task effort prediction models. *IEEE Transactions on Software Engineering*, Vol. 21, No. 8, pp. 674–681, 1995.
- [31] M. Jørgensen and K. Moløkken Østfold. Reasons for software effort estimation error: Impact of respondent role, information collection approach and data analysis method. *IEEE Transaction on Software Engineering*, Vol. 30, No. 12, pp. 993–1007, 2004.
- [32] P. Sentas, L. Angelis, I. Stamelos and G. Bleris. Software productivity and effort prediction with ordinal regression. *Information and Software Technology*, Vol. 47, No. 1, pp. 17–19, 2005.
- [33] M. Shepperd and C.Schofield. Estimating software project effort using analogies. *IEEE Transaction on Software Engineering*, Vol. 23, No. 12, pp. 736–743, 1997.
- [34] K. Srinivasan and D. Fisher. Machine learning approaches to estimation software development effort. *IEEE Transaction on Software Engineering*, Vol. 21, No. 2, pp. 126–137, 1995.

- [35] Bournemouth University. Angel project. <http://dec.bournemouth.ac.uk/ESERG/ANGEL/>. M.Shepperd.
- [36] I. Walkerden and R. Jeffery. An empirical study of analogy-based software effort estimation. *Empirical Software Engineering*, Vol. 4, No. 2, pp. 135–158, 1999.
- [37] C. Walston and C. Felix. A method of programming measurement and estimation. *IBM Systems Journal*, Vol. 1, No. 1, pp. 54–73, 1977.
- [38] 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. 1155–1164, 2005.
- [39] 経済産業省商務情報政策局情報処理振興課. 「情報システムの信頼性向上のための取引慣行・契約に関する研究会」報告書 - モデル取引・契約書. 2007.
- [40] 柿元健, 角田雅照, 大杉直樹, 門田 暁人松本 健一. 協調フィルタリングによる工数見積もり手法におけるデータ数と見積もり精度の関係の分析. *ソフトウェア工学の基礎 XII 日本ソフトウェア科学会 (FOSE2005)*, pp. 77–86, 2005.
- [41] 原田晃, 幕田行雄, 石川貞裕, 大野治, 楠本真二, 井上克郎. ファンクションポイント法を応用した早期見積り技法の提案とそのシステム化. *電子情報通信学会論文誌*, Vol. J89-D, No. 4, pp. 755–766, 2006.
- [42] 戸田航史. 工数見積りモデルの精度を低下させるソフトウェアプロジェクトの特徴分析. Master's thesis, 奈良先端科学技術大学院大学 情報科学研究科情報システム学専攻, 2006.
- [43] 戸田航史, 角田雅照, 門田暁人, 松本健一. 工数見積りモデルで予測できないソフトウェアプロジェクトの特徴分析. *電子情報通信学会技術研究報告. SS, ソフトウェアサイエンス*, Vol. 105, No. 491, pp. 67–72, 2005.

- [44] 本間周二. ”CSK に見る FP 法の実践 - プロジェクト定量化で品質と見積り精度向上”. Technical Report 7, 日経 IT プロフェッショナル, 2004.
- [45] 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 2008 . 日経 BP 社, 2008.
- [46] 瀧進也, 戸田航史, 門田暁人, 柿元健, 角田雅照, 大杉直樹, 松本健一. プロジェクト類似性に基づく工数見積りに適した変数選択法. 情報処理学会論文誌, Vol. 49, No. 7, pp. 2338–2348, 2008.
- [47] 渡邊瑞穂, 柿元健, 戸田航史, 門田暁人, 松本健一. ソフトウェア開発工数予測における特異プロジェクト除去の効果. 日本ソフトウェア科学会第 24 回大会論文集 CD-ROM(講演番号 7B-2), September 2007.
- [48] 奥野忠一, 久米均, 芳賀敏郎, 吉澤正. 多変量解析法. 日科技連, 1971.

付録 A.

2 値変数変換手順

名義尺度，順序尺度の 2 値変数への変換の説明を，開発言語を例にとって行う．

開発言語の中に「COBOL」「C」の 2 つが含まれている場合，説明変数 x は言語が COBOL のときには 1，言語が C のときには 0 を取る．同様に言語が「COBOL」「C」「JAVA」のときにはその自由度 $3-1=2$ に対応して 2 つの定義変数 x_1, x_2 を導入し，その値は以下の表のようになる．

開発言語	x_1	x_2
COBOL	1	0
C	0	1
Java	0	0

これが名義尺度，順序尺度を 2 値変数に変換する手続きである．

付録 B.

; Step 1 ... フィットデータセット候補群の構築

過去プロジェクトの集合を $P(p_1, \dots, p_{pmax})$ とし，見積り対象プロジェクトを $p_t \notin P$ とする．

プロジェクト特性の集合を V とし， V のべき集合の部分集合 $S = \{X | X \in \wp(V) \wedge X \neq \phi \wedge |X| \leq m_{max}\}$ とする．ここで， m_{max} は，見積り対象プロジェクトと一致する特性の最大値である．

for 全ての $s_i \in S$ について {

 フィットデータ候補となるプロジェクト集合

$$F_i = \{p | (v \in s_i) R(p, v) = R(p_t, v)\}$$

 を作成する．

 ここで， $R(p, v)$ はプロジェクト p の特性 v の値をあらわす．

}

; Step 2 ... 見積りモデル群の構築

見積りモデルを構築する上で必要な最小プロジェクト数を $min_projects$ とする .

for 全ての F_i について {

if $|F_i| \geq min_projects$ ならば , 見積りモデル m_i をプロジェクト集合 F_i を用いて構築する .

}

見積りモデル m_{all} を全プロジェクト集合 P を用いて構築する .

Step 2 で構築された全ての見積りモデルの集合を M とする .

; Step 3 ... 最良のフィットデータおよびモデルの選定

モデル m の適合度を $GoF(m)$ とする .

全ての $m \in M$ について $GoF(m)$ を算出し , 最良の $GoF(m)$ を持つモデル m_s およびそのフィットデータ F_s を選定する .

; Step 4 ... モデルの適用

Step 3 で選定された最良のモデル m_s を見積り対象プロジェクト p_t に適用し , 見積り結果を得る .