

リスク管理のためのアンケート回答のクラスタ分析と 混乱プロジェクト発見への応用

濱崎 考成[†], 水野 修[†], 菊野 亨[†], 高木 徳生^{†‡}

[†] 大阪大学 大学院情報科学研究科 情報システム工学専攻

E-mail: {hamasaki, o-mizuno, kikuno}@ist.osaka-u.ac.jp

[‡] オムロン株式会社

E-mail: taka@biwa.kusatsu.omron.co.jp

Abstract

本報告では、リスク管理のために開発したプロジェクト質問表の回答のクラスタ分析、及びソフトウェア開発プロジェクトのクラスタ分析を用いた混乱予測について述べる。

ソフトウェアの開発現場では、プロジェクト管理の重要性が益々高まってきている。特にプロジェクトが混乱するか否かを早い段階で発見することが求められている。我々は、ある企業の協力を得て、ソフトウェア開発プロジェクトのマネージャからのリスク調査アンケート結果に対してロジスティック回帰分析を行い、プロジェクトが混乱するかどうかの予測を行う手法を提案してきた。

しかし、従来の手法ではアンケート項目の一部のみがモデルのパラメータとして利用されるため、ある特定のリスク要因のみが混乱プロジェクトを決定していた。そのため、多くのアンケート項目を活かして、しかも現場に適用可能な手法の開発が強く求められていた。

そこで、本研究では全てのアンケート項目を利用しながらプロジェクトの混乱予測を行う手法の開発を目指す。まずリスク調査アンケートの結果にクラスタ分析を適用することで、プロジェクトを成功プロジェクトと混乱プロジェクトに分類できることを示す。次に、クラスタ分析による分類を利用して、新規プロジェクトの混乱予測を行う手法を提案する。実際のプロジェクトデータを用いた適用実験の結果、実験対象のプロジェクト全てを提案法で正しく予測できることを確認した。

1 はじめに

ソフトウェアの開発期間が次第に短くなり、一方で、高信頼性を求められるようになってきている。こうした状況下で、ソフトウェア開発プロジェクトが抱える問題点を早期に発見することの重要性が高まってきている。そのため、ソフトウェア開発プロジェクトにおけるリスクを

分類したり、リスクの回避を行う技術の開発が望まれている。

ソフトウェア開発プロジェクトにおけるリスクの分類はBoehm [2] らによって行われてきている。また、リスクを分析してプロジェクトの最終状態に関する予測を行う研究も行われている [1, 5]。

我々は、ソフトウェア開発プロジェクトの早期に行うリスク調査アンケートによって、そのプロジェクトが最終的に混乱状態に陥るかどうかを判定する手法を提案してきた [7, 10]。この手法では、過去に収集したアンケートデータに対して統計的分析を行い、ロジスティック曲線に基づくモデルを作成した。そして、新規に入力されるアンケートをこのモデルに当てはめることにより、プロジェクトの最終状態をある程度予測できることを示してきた。また、実際の開発現場への適用を想定して、アンケート調査表の配布・回収やデータの集計、ロジスティックモデルの作成といった一連の作業を支援するシステムを開発した。

しかし、このロジスティック回帰分析に基づく混乱予測モデルでは少数のアンケート項目のみを用いて混乱予測を行っている。これは、プロジェクトを混乱状態に陥らせる特に強いリスク要因を統計的に見つけ、そのリスク要因だけを用了混乱予測を行っているためである。しかし一般にはその少数のリスク要因だけがプロジェクトを混乱させるのではなく、他のリスク要因もプロジェクトが混乱するか否かに影響を与えている。そこで、本研究では特定のリスク要因に限定するのではなく、アンケート結果全体から混乱予測を行う手法を提案することを目的とする。

本研究では、アンケート調査で得られた全ての回答にクラスタ分析を適用してプロジェクトを成功プロジェクトと混乱プロジェクトに分類することを試みる。クラスタ分析では、ある属性を持つデータの集合を属性間の距離に基づいて分類する。その結果、クラスタ分析によりソフトウェア開発プロジェクトを成功プロジェクトと混乱

プロジェクトに分類できることを確認した。また、クラスタ分析を利用したソフトウェア開発プロジェクトの混乱予測の手法を提案する。提案する手法では、クラスタ分析により過去に行われたプロジェクトを成功プロジェクト群と混乱プロジェクト群に分類しておく。次に、新規プロジェクトがどちらのプロジェクト群に類似しているかを計算することで混乱予測を行う。最後に、実際のプロジェクトデータを用いた適用実験を行い、その結果を評価した。

本報告の以降の構成は次のようになっている。まず2節では準備として混乱プロジェクトの定義を与えた後、リスク調査アンケートの説明を行う。3節ではリスク調査アンケートにクラスタ分析を適用して、ソフトウェア開発プロジェクトを成功と混乱に分類する。また4節ではクラスタ分析を応用してプロジェクトの混乱予測を行う手法について述べる。

2 準備

2.1 研究の経過

実際のソフトウェア開発現場では、稀にはあるがプロジェクトの進行状況すら誰にも把握できなくなってしまうようなことが発生する。このようなプロジェクトでは、多くの場合、「混乱状態」と呼ばれる状況に陥ることを繰り返した後に最悪の事態にまで発展してしまう。混乱状態に陥ったプロジェクトのすべてが最悪の事態を招くわけではないが、こうした混乱状態を事前に予測できれば、最悪の事態を防ぐことができる。

これまでソフトウェア開発プロジェクトの混乱状態を予測するためにいくつかの研究が行われてきた。これらの研究を通じて「開発現場のプロジェクトマネージャは無意識にはあるが、プロジェクトの混乱につながるリスク要因をつかんでいる」ということが分かってきた。

そこで、我々はプロジェクトマネージャへのアンケート調査(表1)を実施し、アンケート調査に基づいて、プロジェクトの混乱とリスク要因との間に統計的モデルを作成し、そのモデルを適用して新規プロジェクトの混乱状態を予測する手法を提案してきた[7, 10]。

2.2 混乱プロジェクト

開発現場の状況が制御不可能になってしまうプロジェクトを混乱プロジェクトと呼ぶ[9]。しかし、これでは漠然としているので、ここでは混乱プロジェクトを次のように定義する。

基本的には開発コストと開発期間の2つについて開発計画作成時の推定値と実績値のずれを見て、いずれかが基準値を超えていれば、それは混乱プロジェクトであると

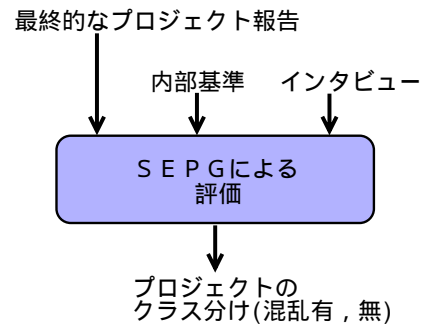


図1. 混乱プロジェクトの分類

判断する。しかし、これでも必ずしも十分ではないので、プロジェクト管理者にインタビューを行って同意が得られたものだけを最終的に混乱プロジェクトと定めることにする(図1)。

2.3 リスク調査アンケート

ここでは過去の研究[7, 10]において作成したリスク調査アンケート(表1)について説明する。

リスク調査アンケートは、プロジェクトマネージャが無意識につかんでいると思われるリスク要因を引き出す目的で作成された。リスク調査アンケートの設計にあたっては、リスク管理に関する専門書や論文[3, 4, 6]と協力企業における内部規約を調査した。その結果、混乱プロジェクトを引き起こすリスク要因を次の5つに整理した。

- (1) 要求仕様に関する問題点
- (2) 見積りに関する問題点
- (3) プロジェクト体制に関する問題点
- (4) 工程計画に関する問題点
- (5) 進捗管理に関する問題点

これらはそれぞれより詳細なレベルの項目に展開されている。例えば(1)の要求仕様に関する問題点は、(1.1)要件が不明確なままでの要求、(1.2)要件の引出し不足、(1.3)要件の理解不足、(1.4)要件に対する顧客側、実現側相互の合意不足、と詳細化されている。なお、これらの調査項目は設計終了時点で全て記入可能であることを目安にしている。

評価についてはプロジェクト管理者が“ある”、“どちらでもない”、“ない”、“分からない”のいずれかを記入する。なお、本研究ではリスク調査アンケートへの回答を分析す

表 1. リスク調査アンケート

1. 要求仕様	
1.1	要件が不明確なままでの要求
1.2	要件の引出し不足
1.3	要件の理解不足
1.4	要件に対する顧客側、実現側相互の合意不足
2. 見積り	
2.1	見積りの項目不足
2.2	見積りの重要さの認識不足
2.3	非技術的圧力に妥協
2.4	技術的課題に対する楽観的過ぎる見積り
2.5	見積りの根拠不足
3. プロジェクト体制	
3.1	要因のスキル不足
3.2	プロジェクト体制整備の必要性認識不足
4. 工程計画	
4.1	作業に対する責任分担不明確
4.2	作業成果物定義不十分
4.3	マイルストーンやレビュー時期の設定不足
4.4	計画に対する関係者全員のコミットメントなし
4.5	計画に対する上級マネージャのレビュー不足
4.6	進捗管理方法不明確
5. 進捗管理	
5.1	技術的側面でのリスク管理不足
5.2	モラル不足
5.3	工数不足
5.4	要件、仕様変更管理不足
5.5	進捗状況報告不足
5.6	進捗管理データ不足

るための基礎データとするために、“ある”、“どちらでもない”、“ない”という回答に評価値 3, 2, 0 を対応づける。また“分からない”という回答の判断については、SEPG や現場との協議を通じて割り当てる値を決定した。すなわち、“分からない”と回答した項目については、そうしたリスク要因はプロジェクト中で見受けられなかったが“ない”とは言い切れない要因が存在した、と解釈して評価値 1 を対応づける。

2.4 従来の混乱予測手法 [7,10]

従来提案している手法 [7, 10] では、2.3 節で説明したリスク調査アンケートの回答結果に対し、ロジスティック回帰分析を行うことで 2.2 節で述べたプロジェクトの「混乱」と「成功」を分類、予測していた。

ロジスティック回帰モデルによる混乱予測では、プロジェクトを混乱状態に陥らせる特に強いリスク要因を統計的に見つけ、そのリスク要因だけを用いた混乱予測を行う。そこでロジスティック回帰モデルを作成するに当たっては、アンケートの各項目をモデルの変数の候補とし、ステップワイズ法に基づいて統計的に有意な変数のみを選択していた。その結果、リスク要因として使われる変数は強い有意性を示した一部の変数のみとなってしまう、ほとんどのアンケート結果は混乱予測に寄与していなかった。

しかし一般にはその少数のリスク要因だけがプロジェクトを混乱させるのではなく、他のリスク要因もプロジェクトが混乱するか否かに影響を与えていると考えられる。そこで、本研究ではアンケートの各項目への回答結果の統計的な有意性に着目するのではなく、回答傾向の類似性に着目し、混乱プロジェクトを予測する手法を提案する。

3 アンケート回答のクラスタ分析

3.1 クラスタ分析手法

クラスタ分析では、個体間の距離¹を基に、類似している個体同士を集めてクラスタ (類似なものから構成される部分集団) を作ることによって、個体の分類を行う。

本研究では、リスク調査アンケートの類似度を基にソフトウェア開発プロジェクトを成功プロジェクトと混乱プロジェクトの 2 つのクラスタに分類する。クラスタ分析の手順は以下ようになる [11, 12]。

1. 初期状態では、1 つの個体が 1 つのクラスタをなしているとする。
2. クラスタ間の距離が最も小さい 2 つのクラスタを見つける。
3. 2. で見つけた 2 つのクラスタを 1 つのクラスタに融合する。2 つのクラスタが 1 つにまとめられたので、総クラスタ数が 1 つ減る。
4. クラスタの数が 1 になるまで 2, 3 を繰り返す。

距離としては、ユークリッド距離、重み付きユークリッド距離などがある。ここで対象 α と対象 β の特性値を

$$\mathbf{x}_\alpha = (x_{\alpha 1}, x_{\alpha 2}, \dots, x_{\alpha p})$$

$$\mathbf{x}_\beta = (x_{\beta 1}, x_{\beta 2}, \dots, x_{\beta p})$$

とする。対象 α と対象 β の距離を $d_{\alpha\beta}$ とするとユークリッド平方距離は、

$$d_{\alpha\beta}^2 = \sum_i (x_{\alpha i} - x_{\beta i})^2$$

となり、重み付きユークリッド距離は、

$$d_{\alpha\beta}^2 = \sum_i k_i (x_{\alpha i} - x_{\beta i})^2$$

となる。

クラスタ分析においてはクラスタ間の距離をどう定義するかが重要である。2 つのクラスタ間の距離はそれぞれ

¹距離の他に、類似係数など類似度を表す尺度を用いることもある

表 2. 各手法で用いるパラメータ値

手法	α_p	α_q	β	γ	使用する式
最短距離法	0.5	0.5	0	-0.5	(1)
最長距離法	0.5	0.5	0	0.5	(1)
メディアン法	0.5	0.5	-0.25	0	(1)
重心法	n_p/n_r	n_q/n_r	$-(n_p n_q)/n_r^2$	0	(2)
群平均法	n_p/n_r	n_q/n_r	0	0	(2)
ワード法	$(n_p + n_i)/(n_r + n_i)$	$(n_q + n_i)/(n_r + n_i)$	$-n_i/(n_r + n_i)$	0	(2)

のクラスタを何らかの意味で代表する点の間の距離として定義できる。各クラスタが複数個の個体を含んでいる場合には、その代表点をどう決めるかが問題となる。本研究で使用した手法について説明する。

最短距離法 2つのクラスタに含まれる個体の組み合わせのうち、最も近い個体間の距離をクラスタ間の距離とする。

最長距離法 2つのクラスタに含まれる個体の組み合わせのうち、最も遠い個体間の距離をクラスタ間の距離とする。

メディアン法 2つのクラスタに含まれる個体の組み合わせのうち、最も遠い個体間の距離と最も近い個体間の距離の中間の値をクラスタ間の距離とする。

重心法 各クラスタに含まれる個体の個数も考慮して、重心間の距離をクラスタ間の距離とする。

群平均法 2つのクラスタに含まれる個体間の組み合わせの距離の平均値をクラスタ間の距離とする。

ワード法 2つのクラスタを結合してできるクラスタ内の変動は、結合する前の2つのクラスタのそれぞれの変動の和より増加するが、この変動の増加分を、クラスタ間の距離とする [11]。

これらの手法によって、クラスタ p とクラスタ q が結合してクラスタ r ができると仮定する。クラスタ r と他のクラスタ i の距離 d_{ir} は、結合する前の距離 d_{ip}, d_{iq} を用いて、次の式 (1), (2) のように計算される。この式に含まれる4つのパラメータ $\alpha_p, \alpha_q, \beta, \gamma$ の値を変えることでそれぞれの手法が定義される。表 2 に各手法で利用されるパラメータの一覧を示す。

$$d_{ir} = \alpha_p d_{ip} + \alpha_q d_{iq} + \beta d_{pq} + \gamma |d_{ip} - d_{iq}| \quad (1)$$

$$d_{ir}^2 = \alpha_p d_{ip}^2 + \alpha_q d_{iq}^2 + \beta d_{pq}^2 + \gamma |d_{ip}^2 - d_{iq}^2| \quad (2)$$

表 3. アンケート結果

プロジェクト	Q1.1	Q1.2	Q1.3	...	Q5.6
PJ1	0	0	0	...	0
PJ2	0	0	0	...	0
PJ3	0	0	0	...	0
PJ4	3	3	2	...	0
PJ5	0	0	0	...	0
PJ6	0	3	2	...	2
PJ7	0	0	2	...	0
PJ8	0	2	3	...	0
PJ9	0	2	0	...	2
PJ10	2	2	0	...	3
:	:	:	:	:	:
PJ40	2	2	2	...	2

3.2 分析対象データ

あるソフトウェア開発企業の協力を得て、1996年から1998年までに開発された40プロジェクト(PJ1~PJ40)のデータを分析対象とする。なお、これらは全て開発が終了しており、混乱したかどうかの判定もすでにSEPGによって行われている。表1のリスク調査アンケートを行ったところ、表3に示す結果を得た。このアンケートの回答内容を各プロジェクトの特性値とする。

3.3 クラスタ分析の結果

まず、クラスタ間の距離はユークリッド距離を用いることにする。次に、クラスタ分析手法としては最短距離法、最長距離法、群平均法、重心法、メディアン法、ワード法の6つを適用する。

最長距離法を用いた分析で得られた樹形図を図2に示す。樹形図では、左側に分析対象の40個のプロジェクト(PJ1~PJ40)が並んでいる。ここで(c)と書かれたプロジェクトは混乱プロジェクトである。樹形図は最初に各

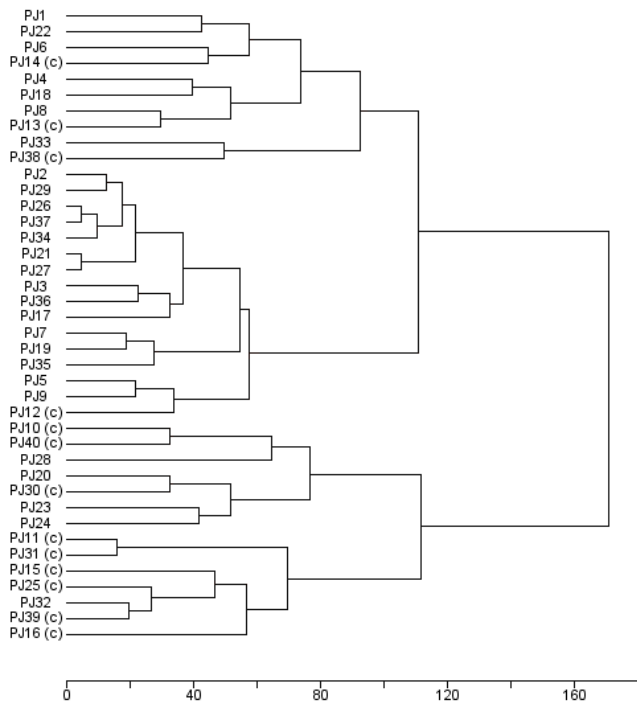


図 2. 最長距離法による樹形図

表 4. 最長距離法による分類

	クラスタ 1	クラスタ 2	計
成功プロジェクト	22	5	27
混乱プロジェクト	4	9	13
計	26	14	40

プロジェクトが1つのクラスタであると見なして、クラスタが結合されていく過程を表している。

この樹形図で全体が1つのクラスタにまとまる直前に注目し、40個のプロジェクトを2つのクラスタに分ける。それぞれをクラスタ1、クラスタ2と呼ぶ。表4にクラスタ1と2に含まれる成功プロジェクトと混乱プロジェクトの数を示す。

表4より、クラスタ1が成功プロジェクト22個、混乱プロジェクト4個で構成され、クラスタ2が成功プロジェクト5個、混乱プロジェクト9個で構成されていることがわかる。

3.4 フィッシャーの正確確率検定

次に、このクラスタ分析による分類と実際のプロジェクトの「成功」「混乱」という判定結果の間に統計的な意味で関連があるかを調べる。そのため表4の分割表に対

表 5. 各手法による分類

群平均法		
	クラスタ 1	クラスタ 2
成功プロジェクト	22	5
混乱プロジェクト	2	11
計	24	16

p 値: 0.000104

ワード法		
	クラスタ 1	クラスタ 2
成功プロジェクト	20	7
混乱プロジェクト	9	4
計	29	11

p 値: 0.0154

重心法		
	クラスタ 1	クラスタ 2
成功プロジェクト	27	0
混乱プロジェクト	11	2
計	38	2

p 値: 0.100

最短距離法		
	クラスタ 1	クラスタ 2
成功プロジェクト	27	0
混乱プロジェクト	12	1
計	39	1

p 値: 0.325

メディアン法		
	クラスタ 1	クラスタ 2
成功プロジェクト	27	0
混乱プロジェクト	12	1
計	39	1

p 値: 0.325

して、フィッシャーの正確確率検定を行った。

帰無仮説: 2つのクラスタ1,2とプロジェクトの「成功」「混乱」の判定結果は独立である。

対立仮説: 2つのクラスタ1,2とプロジェクトの「成功」「混乱」の判定結果は独立でない。

有意水準: 0.05

p 値: 0.00358

その結果、帰無仮説は棄却され、2つのクラスタと、プロジェクトの「成功」「混乱」は独立でないと結論づけられた。つまり2つのクラスタは「成功」プロジェクトと「混乱」プロジェクトを表しているといえる。

他の分析手法についても同様に分析を行った。その結果(分割表、フィッシャーの正確確率検定の p 値)を表5に示す。

表5に示すように、今回分析を行ったデータに対して最長距離法、群平均法、ワード法では有意水準を0.05とした時に統計的に有意な分類結果が得られた。一方、他の手法では有意な分類結果は得られなかった。

一般にクラスタ分析を用いて分類を行う際にどの手法が良いという基準は特に無く、どの手法を用いれば有意な分類になるかは対象としたデータにも依存する。しかしどのようなデータに対してもいずれかの手法を用いることで表4に示すような有意な分類を行うことができると考えられる。そこで次節では有意な分類がなされたクラスタを用いて混乱予測を行う手法を提案する。

4 クラスタ分析による混乱予測

3.3節及び3.4節における分析から、リスク調査アンケートの結果からソフトウェア開発プロジェクトを成功プロジェクトと混乱プロジェクトの2つのクラスタに分類できることが分かった。この2つのクラスタはともにプロジェクトの集合であるから、各クラスタに含まれるプロジェクトと新規プロジェクトを比較することも可能である[8]。そこで、クラスタ分析による分類結果を利用して、新規プロジェクトの混乱予測を行う。

4.1 混乱予測の手順

提案する混乱予測の手順は次の通りである。

step1 予測を行うための予測元となるプロジェクトのデータを集める。

step2 リスク調査アンケートの結果をクラスタ分析して予測元プロジェクトを2つに分類する。分類で得られた2つのクラスタを、成功プロジェクトクラスタと混乱プロジェクトクラスタと呼ぶ。

step3 成功プロジェクトクラスタ、混乱プロジェクトクラスタと予測対象プロジェクトの距離をそれぞれ求め、予測を行う。

以降では、各stepについての詳細な説明を行う。

step1 予測を行うための予測元プロジェクトを集める。

予測元プロジェクトとなるのは、過去にリスク調査アンケートを行ったプロジェクトで、プロジェクトの結果(成功、混乱)が評価されているプロジェクトである。予測元プロジェクトとしてはある程度の数のプロジェクトが必要である。

step2 予測元プロジェクトをリスク調査アンケートの結果を基に、クラスタ分析を用いて分類する。

分析手法としては、成功プロジェクトと混乱プロジェクトを有意に分類できた分析手法だけを用いることにする。その分析結果から、成功プロジェクトクラスタと混乱プロジェクトクラスタを得る。

step3 リスク調査アンケートを行った新規の予測対象プロジェクトの混乱予測を行う。

予測対象プロジェクトを1つのクラスタと見なして、成功プロジェクトクラスタとの距離(D_s とする)、混乱プロジェクトクラスタとの距離(D_c とする)をそれぞれ求める。

もし $D_s < D_c$ なら予測結果は「成功」とし、 $D_s \geq D_c$ なら予測結果は「混乱」とする。

4.2 混乱予測の実験的評価

3.2節で述べた40個のプロジェクトのうち、1996年、1997年に行われた32個のプロジェクト(PJ1~PJ32)を予測元プロジェクトとする。予測対象プロジェクトとしては1998年に行われた8個のプロジェクト(PJ33~PJ40)を用いる。

予測元プロジェクトに対して、ウォード法を用いてクラスタ分析を行った結果を図3と表6に示す。

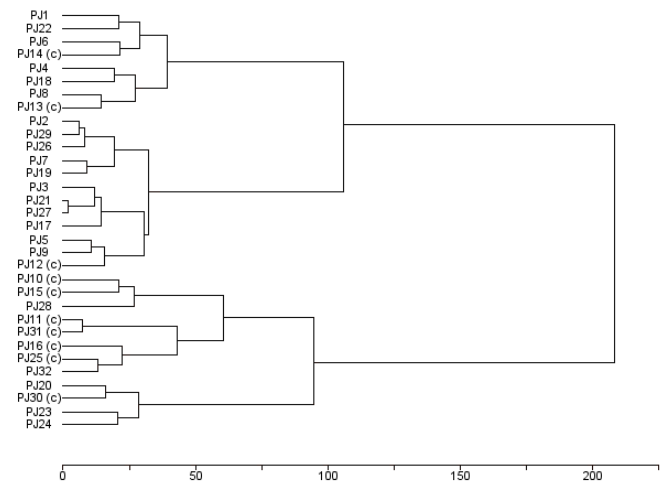


図3. 予測元プロジェクトの樹形図(ウォード法)

表6. 予測元プロジェクトの分類(ウォード法)

	クラスタ1	クラスタ2	計
成功	17	5	22
混乱	3	7	10
計	20	12	32

p 値 : 0.0184

図3と表6より、有意な分類を得ることができたと言える。ここで、それぞれのクラスタに含まれる成功プロジェ

表 7. 予測元プロジェクトの分類 (最長距離法, 群平均法)

最長距離法		
	成功クラス	混乱クラス
成功	22	0
混乱	7	3
計	29	3

p 値 : 0.0242

群平均法		
	成功クラス	混乱クラス
成功	18	4
混乱	4	6
計	22	10

p 値 : 0.0369

クトと混乱プロジェクトの比率から、クラス 1 を「成功」クラス、クラス 2 を「混乱」クラスとする。

同様に、最長距離法、群平均法の 2 つの手法で有意な分類結果が得られた (表 7)。なお、その他の手法では有意な分類結果を得ることができなかった。以降の予測実験では最長距離、群平均、ワードの 3 手法のみを用いる。

次に、1998 年に行われた 8 個のプロジェクト (PJ33 ~ PJ40) の混乱予測をこれら 3 つの手法を用いて行った。提案法による予測結果を表 8 に示す。表中で*印を付けたプロジェクトは実際に混乱したことを表す²。また、予測結果には D_c/D_s の値も併記する。 $D_c/D_s > 1$ であるときプロジェクトは成功する予測し、 $D_c/D_s < 1$ であるとき混乱と予測する。

実験の結果、予測に用いた 3 つの手法のいずれについても、8 つのプロジェクトの成功と混乱を正確に予測できることを確認した。

4.3 考察

表 8 に示す各手法の予測結果については、次のような特徴が見られた。

ワード法 PJ33 ~ 37 (実際には成功したプロジェクト) については明確に成功クラスに分類できるものがある一方で、PJ35 のように $D_c/D_s = 1.09$ となり、辛うじて成功に分類されたものもある。また、PJ38 ~ PJ40 (実際に混乱したプロジェクト) については D_c/D_s の値は 0.64 ~

²PJ33 から PJ40 は実際にはすべて開発が終了しているので、混乱したかどうかの判定もすでに SEPG によって行われている。

0.88 となっており、混乱クラスへの分類はうまくいっている。

最長距離法 平均的に見て、 D_c/D_s の値があまり散らばらず、特に PJ33 と PJ35 については、それぞれ 1.07、1.05 と非常に 1 に近い値となっており、成功と混乱の境界が曖昧である。ただし、PJ38 ~ PJ40 については、 D_c/D_s の値は 0.51 ~ 0.74 となっており、明確に混乱クラスへと分類できている。

群平均法 PJ33 ~ 37 については、 D_c/D_s の値が 1.30 ~ 2.44 と比較的大きくなっており、成功クラスへの分類ははっきりしている。PJ38 ~ 40 についても、 D_c/D_s の値は 0.54 ~ 0.87 となっており、成功クラスと混乱クラスの区別が明確であることが分かる。

以上の結果から、本研究で実験を行ったデータに対してではあるが、群平均法を用いたクラスタリングがもっとも成功クラスと混乱クラスの特徴をつかんだ分類を行えたと考えられる。

5 まとめと今後の課題

本報告では、プロジェクトマネージャに対するリスク調査アンケートの結果にクラスタ分析を適用することでソフトウェア開発プロジェクトを成功プロジェクトと混乱プロジェクトに分類できることを確認した。次に、クラスタ分析によって得られた成功プロジェクトクラスと混乱プロジェクトクラスを利用して、新規プロジェクトの混乱予測を行う手法を提案した。更に、その手法によって高い精度で混乱予測が可能であることを確認した。

今後の課題としては、より多くのソフトウェア開発プロジェクトに対して混乱予測を行い提案手法の有効性の検証を行うことがある。そして、クラスタ分析の手法のうち、どれが最も有効であるかを調べる必要がある。また、混乱すると予測した場合にそれを回避する方法についての研究も進めていく必要がある。

謝辞

本研究を行うに当たっては、群馬大学社会情報学部の青木繁伸教授による Web 上の統計分析ツール (<http://aoki2.si.gunma-u.ac.jp/lecture/index.html>) を利用させて頂きました。大変有用なツールを Web で公開して頂いていることに、この場を借りて感謝致します。

参考文献

- [1] T. R. Adler, J. G. Leonard, and R. K. Nordgren: "Improving risk management: moving from risk elimi-

表 8. 提案法による予測結果

プロジェクト	ワード法			最長距離法			群平均法		
	D_s	D_c	予測結果 (D_c/D_s)	D_s	D_c	予測結果 (D_c/D_s)	D_s	D_c	予測結果 (D_c/D_s)
PJ33	33.4	39.3	成功 (1.18)	96	103	成功 (1.07)	57.7	75.1	成功 (1.30)
PJ34	14.6	32.3	成功 (2.21)	108	169	成功 (1.56)	39.8	92.7	成功 (2.33)
PJ35	17.4	18.9	成功 (1.09)	112	118	成功 (1.05)	46.5	68.5	成功 (1.47)
PJ36	15.6	26.5	成功 (1.70)	95	137	成功 (1.44)	41.1	83.5	成功 (2.03)
PJ37	10.3	27.3	成功 (2.64)	109	162	成功 (1.48)	37.0	90.1	成功 (2.44)
PJ38*	57.6	50.6	混乱 (0.88)	110	81	混乱 (0.74)	83.1	70.5	混乱 (0.85)
PJ39*	50.9	32.5	混乱 (0.64)	103	53	混乱 (0.51)	82.9	44.5	混乱 (0.54)
PJ40*	54.4	40.0	混乱 (0.74)	106	77	混乱 (0.72)	74.6	65.1	混乱 (0.87)

nation to risk avoidance,” Information and Software Technology, Vol.41, pp.29-34, 1999.

[12] 古谷野 亘: 多変量解析ガイド, 川島書店, 1988.

- [2] B. W. Boehm: “Industrial software metrics top 10 list,” IEEE Software, Vol.4, No.5, pp.84-85, 1987.
- [3] E. H. Conrow and P. S. Shishido: “Implementing risk management on software intensive project,” IEEE Software, Vol.14, No.3, pp.83-89, 1997.
- [4] R. Fairley and P. Rook: “Risk management for software development,” In Software Engineering, IEEE CS Press, pp.387-400, 1997.
- [5] J. Jiang and G. Klein: “Software development risks to project effectiveness,” Journal of Systems and Software, Vol.52, pp.3-10, 2000.
- [6] D. W. Karolak: *Software Engineering Risk Management*, IEEE CS Press, CA, 1996.
- [7] O. Mizuno, T. Kikuno, Y. Takagi, and K. Sakamoto: “Characterization of Risky Projects based on Project Managers’ Evaluation,” Proc. of 22nd International Conference on Software Engineering(ICSE2000), pp.387-395, 2000.
- [8] H. C. Romesburs 著, 西田 英郎, 佐藤 嗣二 訳: 実例 クラスタ分析, 内田老鶴圃, 1992.
- [9] E. Yourdon: *Death March : The Complete Software Developer’s Guide to Surviving ‘Mission Impossible’ Projects*, Prentice Hall Computer Books, 1997.
- [10] 足立卓也, 水野修, 菊野亨, 高木徳生, 坂本啓司: ”アンケート調査に基づく開発中のプロジェクトの混乱予測とその予測作業支援システムの開発,”ソフトウェアシンポジウム 2000 論文集, pp.146-153, 2000.
- [11] 奥野 忠一, 芳賀 敏郎, 矢島 敬二, 奥野 千恵子, 橋本 茂司, 古河 陽子: 続多変量解析, 日科技連出版社, 1976.