



---

# 卒業研究報告書

平成30年度

---

研究題目

クラスタリングを用いた作業履歴の目的分類

---

指導教員 上野秀剛 准教授

---

氏名 鷺尾 巧

---

平成31年1月17日 提出

奈良工業高等専門学校 情報工学科

# クラスタリングを用いた作業履歴の目的分類 1 上野

研究室 鷲尾 巧

ソフトウェア開発者の能力向上や開発プロセスの改善をするための手法である Personal Software Process が提唱されている。PSPはソフトウェア開発者の作業履歴を記録し、実装やテスト、設計、会議といった個々の開発工程にどれだけの時間を費やしているかを分析し、効率の改善やプロセスの改善に役立てる手法である。そこで、PSPでの作業履歴計測を支援するためのツールである TaskPit がある。先行研究で大橋らは TaskPit によって計測された作業履歴を用いてその前後の情報からその作業がどういった目的で行われているかを推定する手法を提案した。これまで行われていた手法では何日分かの作業を先に計測して機械学習で学習させてから開発作業の自動推定を行っていた。事前に学習が必要となると開発作業の途中でのプロセスの改善や開発作業の初期の段階での PSP が行えない。そこで本研究ではクラスタリングを用いてあるアプリケーションを使用している場合の作業名と作業時間、打鍵数、左右クリック数などをもとに作業目的が同一の作業群を推定する手法を提案する。クラスタリングは教師なしの機械学習アルゴリズムなので、クラスタリングを用いることで事前学習なしで作業目的を推定することができる。クラスタリングを用いて作業を行っている目的（実装や設計）をクラスタごとに分類し、後作業として各作業に作業目的をラベル付けする手法を提案する。さらに、クラスタリングでは計測される作業量（クリック数や打鍵数）のうち、どれが分類に重要であるかを確認した。提案手法では、大橋らを使用したデータと同じデータを使用する。その計測された作業履歴から、作業の順序関係の特徴として用いることができるように処理を施し、文字列データをクラスタリングでも使えるようにダミー変数を用いる。最後に、処理を施した作業履歴データをクラスタリングに使う。クラスタリングを行ったデータと正解データを照らし合わせて各クラスタにどの程度ばらつきがあるかをエントロピーを用いて確認する。実験は、5名の被験者の5日分の作業履歴を用いた。各被験者ごとにクラスタに分類し、各クラスタのエントロピーをだし、次にそのエントロピーすべての平均をだし、最後に被験者全員のエントロピーの平均をだして評価する。

実験の結果、右クリック数を分類に用いないほうが分類精度が上がることが分かった。また、クラスタ数をふやすことで後作業が増えるが分類の精度を上げることができた。

# 目次

<b>1</b>	<b>はじめに</b>	<b>2</b>
<b>2</b>	<b>PSP支援システム</b>	<b>3</b>
<b>3</b>	<b>準備</b>	<b>4</b>
3.1	TaskPit . . . . .	4
3.2	Task . . . . .	4
3.3	Aim . . . . .	5
3.4	時系列情報を用いたAimの予測 . . . . .	6
3.5	クラスタリング . . . . .	7
<b>4</b>	<b>実験</b>	<b>9</b>
4.1	提案手法 . . . . .	9
4.1.1	パラメータ . . . . .	9
4.1.2	処理手順 . . . . .	9
4.2	対象データ . . . . .	10
4.3	評価 . . . . .	10
<b>5</b>	<b>結果と考察</b>	<b>13</b>
5.1	パラメータごとの平均エントロピー . . . . .	13
5.2	被験者ごとのエントロピー . . . . .	14
5.3	クラスタ数 . . . . .	15
5.4	考察 . . . . .	16
<b>6</b>	<b>おわりに</b>	<b>17</b>
	謝辞	18
	参考文献	19

# 1 はじめに

ソフトウェア開発者の能力向上やプロセス改善に用いる手法として Personal Software Process(PSP)[1] が提唱されている。PSPはソフトウェア開発者の作業履歴を記録し、実装やテスト、設計、会議といった個々の開発工程にどれだけの時間を費やしているかを分析し、効率の改善やプロセスの改善に役立てる手法である。

これまでにPSPにおける作業履歴を記録するための支援システムが複数提案されている[2][3][4]。これらのシステムはソフトウェア開発作業における作業名・所要時間などを任意のタイミングで記録することができる。ただし、記録者は現在行っている作業の作業名を手動で入力する必要があり、記録忘れや作業への負担が問題となる。

この問題を解決するシステムとしてTaskPit[5]が存在する。このシステムは、現在使用しているアプリケーション名、あるいはそれに関連付けられた作業名と作業時間、打鍵数、左右クリック数を自動的に記録する。本研究室では以前から作業履歴を用いてその作業がどういった目的で開発作業を行っているかを自動で推定する研究が行われていた[6]。

これまで行われていた手法では何日分かの作業を先に計測して機械学習で学習させてから開発作業の自動推定を行っていた。事前に学習が必要となると開発作業の途中でのプロセスの改善や開発作業の初期の段階でのPSPが行えない。そこで本研究ではクラスタリングを用いてあるアプリケーションを使用している場合の作業名と作業時間、打鍵数、左右クリック数などをもとに作業目的が同一の作業群を推定する手法を提案する。クラスタリングは教師なしの機械学習アルゴリズムなので、クラスタリングを用いることで事前学習なしで作業目的を推定することができる。クラスタリングを使うことで、PSPに適用したいタイミングに分類されたクラスターをラベル付けすることができ、PSPへの即時適用が可能となる。

## 2 PSP 支援システム

PSPとはソフトウェア開発者が開発プロジェクトを効率的に進めるために費やした時間を計測し、分析することでプロセスを改善する手法である。開発プロジェクトを進める前に各作業に必要な時間を見積もり、計測されたデータとの比較を行うことでプロセスの問題点を明らかにする。問題点を明らかにすることで開発者の見積もりスキルを向上させ、効率のよい作業計画ができ、さらにプロジェクトの品質も向上させることができる。

PSPの支援を行うシステムとして作業の時間を自動で計測するProcess Dashboard[2], Task Coach[3], Slim Timer[4]などがある。これらのシステムは作業が切り替わったときに手動で次の作業を入力する必要がある。そのため、計測データに漏れが発生したり計測作業に気を取られてプロジェクトに支障をきたす可能性がある。

本研究で使用するシステム、TaskPit[5]は作業の計測が自動化されておりアクティブになっているソフトウェアのウィンドウ名と作業名を関連付けて計測することで作業の切り替わり判別も自動化されている。TaskPitでは作業時間や作業名だけでなく、作業時間や打鍵数、マウスクリック数を同時に計測している。しかし計測された作業履歴には各Taskの目的(Aim)が記録されていない。そのためプロセス改善には各TaskのAimを調査し、どのAimにどの程度時間をかけているのかを分析する必要がある。

先行研究[6]で大橋らはRandom Forestsを用いてTaskの時系列情報からAimを推定する手法を提案している。本研究ではRandom Forests以外の手法でどの程度の推定精度がでるかを調査する。また、計測されたパラメータのうちどれがAimの推定に有用であるかを分析する手法を提案する。

## 3 準備

### 3.1 TaskPit

TaskPit は門田らによって提案された PSP 計測支援システムである。本システムは、あるアプリケーションに対する一連の操作を1つの作業とみなし、各アプリケーション上で行われた操作とその時間を記録することで各作業に費やした時間を計測する。これによって、例えば、Wordを使った「テキスト閲覧・編集」、Eclipseを用いた「コーディング」といった粒度で作業を計測することができる。このようなアプリケーションと作業とのひも付けは、TaskPit の設定ファイルで行うことができる。

また、TaskPit はチームとして計測を行う際、計測者に対してのプライバシーと個人情報保護に配慮しており、キーストロークの内容、アプリケーションの動作内容、ウィンドウ名、ファイル名、ファイルの内容等については記録しない。さらに、TaskPit にあらかじめ設定されていないすべてのアプリケーションの使用履歴を記録することは「監視されている」という感覚を計測者に与え、システム導入の妨げになる恐れがあるため「登録外」として計測する。TaskPit は作業名、作業開始時間、作業終了時間、右クリック回数、左クリック回数、打鍵数を記録し、TaskLog というファイルに保存する。

本研究ではTaskPit が計測する作業をTask、開発者がTaskを行っている目的をAimと定義し、Aim が同一のTask群を推定する。

### 3.2 Task

Taskとは1つのソフトウェアに対する連続したユーザの操作である。TaskPit はアクティブなソフトウェアに対する操作を監視し、連続した操作を1つのTaskとして記録する。Task名はアクティブなソフトウェアの実行ファイル(exe)名とウィンドウ名から識別する。Webブラウザのような複数のTaskで利用されるソフトウェアであっても、ウィンドウタイトルが違えば異なるTaskとして記録することができる。前説で述べたように、TaskPit は作業名、作業開始時間、作業終了時間、右クリック数、左クリック数、打鍵数をTaskLog というファイルに保存する。ここで、打鍵数に関しては、数は記録するが入力内容は記録しない。TaskLog に保存されている内容をそのままTaskとして使用し、同一のAimのTask群を推定するのに用いる。本研究ではある1つのログデータに記録されたすべてのTask  $T$  を連続したTask  $t$  の順序集合で表す。

$$T = \{t_0, t_1, t_2, \dots, t_n\} \quad (1)$$

### 3.3 Aim

AimとはユーザがTaskを行う目的であり，各TaskはAimを達成するために行われる．本研究では，1つのAim Aは連続した順序Task tの部分集合で表す．ある一つのログデータから構成されるTask Tの部分集合がAim Aとなる．

$$A = \{t_m, t_{m+1}, t_{m+2}, \dots, t_k\} \quad (2)$$

また，図1にTaskとAimの関係を示す．図1では各四角を1つのTaskとしてその中の文字はTask名，キーボードとマウスは計測者の相対的な作業量（打鍵数やクリック数）を示す．隣り合う3つのTaskはその上に書いてあるAimを達成するために行われているものである．各TaskはAimの特徴を示している．例えば，Aimが”実装”のときにはソースコード閲覧編集のときに打鍵数が多くなりクリック数が少なく，仕様書閲覧編集や設計書閲覧編集のときには打鍵数が少ない．Aimが”設計”のときにはソースコード閲覧編集というTaskはなく，設計書閲覧編集のときに打鍵数が多くなっている．Aimが”テスト”のときはTask名は仕様書閲覧編集，設計書閲覧編集，ソースコード閲覧編集となっていてAimが”実装”のときと同じだが，ソースコード閲覧編集のときにクリック数多くなっている．このように各AimごとにTaskには特徴がある．

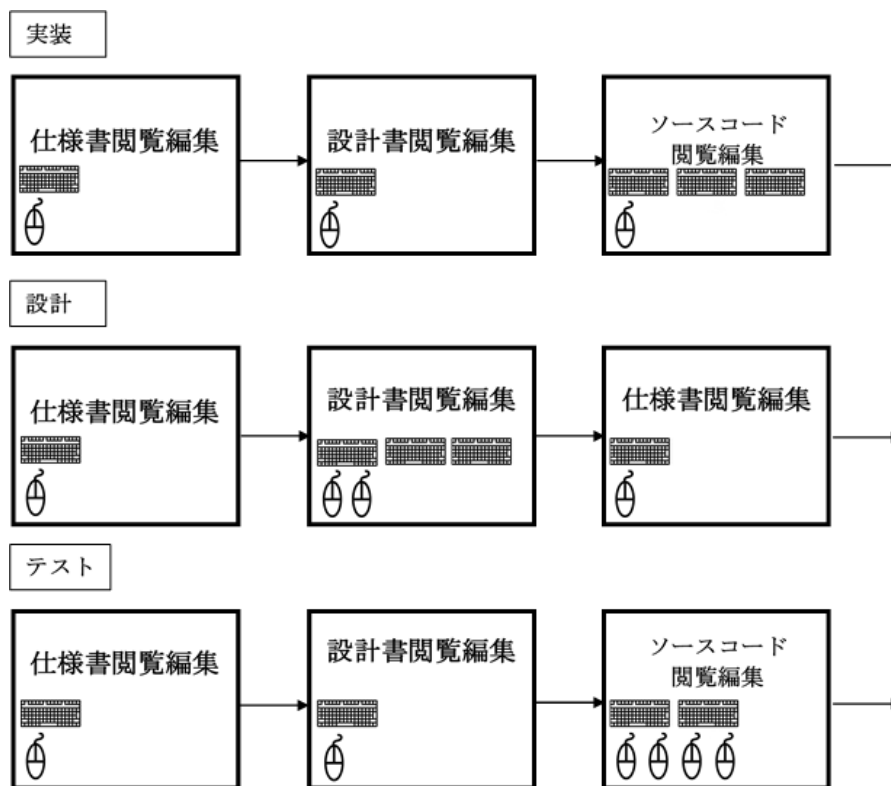


図1 TaskとAimの関係

### 3.4 時系列情報を用いた Aim の予測

ある1つの Aim は連続した複数の Task で構成されており、Task 列には Aim ごとに種類、順序、作業量に特徴がある。Task 名が同じであっても作業量の違いにより複数の Aim に分類されうるため、Task の名前や作業量だけでは Aim の判別は難しい。

そこで、大橋らは時系列情報を用いた Aim の予測をした。Task の名前や作業量に加え順序を特徴として用いることで、前後の Task の中から主要となる Task の情報を使い、Aim の判別ができる。

大橋らは機械学習アルゴリズムの1つである Random Forests を用いて予測をした。計測された Task 列には Aim ごとに Task の種類、順序、作業量（クリック数や打鍵数）に特徴がある。例えば、Aim が”実装”の場合”ソースコード閲覧編集”という打鍵数の多い Task があるが、Aim が”設計”の場合には”ソースコード閲覧編集”という Task はなく、主に打鍵数の多い”設計書閲覧編集”という Task がある。このように Aim が違えば、Task の種類や作業量に特徴がある。しかし、Task の種類によっては同じ Aim でも複数の Aim に分類されてしまう場合がある。そこで大橋らは Task の順序の特徴を使って、前後の Task の中から主要となる Task の情報を使って Aim を推定する手法を提案した。大橋らは推定に用いた特徴として以下の8種類をあげている。

1. 予測対象 Task  $t$  の名前:  $t.Name$
2. 予測対象 Task  $t$  の左クリック数:  $t.LClick$
3. 予測対象 Task  $t$  の右クリック数:  $t.RClick$
4. 予測対象 Task  $t$  の打鍵数:  $t.Key$
5. 予測対象 Task  $t$  の前後の各 Task 名:  $tp - x.Name, \dots, tp.Name, \dots, tp + x.Name$
6. 前後の Task において左、右クリック数、打鍵数それぞれの数値が最も高い Task 名:  
 $t.1stLClick, t.1stRClick, t.1stKey$
7. 前後の Task において左、右クリック数、打鍵数それぞれの数値が二番目に高い Task 名:  
 $t.2ndLClick, t.2ndRClick, t.2ndKey$
8. 前後の Task において左、右クリック数、打鍵数それぞれの数値が三番目に高い Task 名:  
 $t.3rdLClick, t.3rdRClick, t.3rdKey$



これら 8 種類の特徴のうち 6 から 8 は計測された Task を基に特徴として追加することで，予測対象 Task の前後作業の特徴を加味した Aim の判別を行う．前後作業の特徴を用いたときの Aim の推定精度として 95%，用いないときでは 85% という結果が出て，用いることでより高い精度で推定ができることを大橋らは示している．

しかし，大橋らの手法は Random Forests を用いているため，事前の学習のためにデータを計測するとともに，各 Task の Aim を収集する必要がある．例えば，この大橋らの手法だと本来は開発作業の効率化に用いる手法のはずなのに一日の作業終わりにデータを使って Aim を自動推定するといったようにどのタイミングでも自動推定ができるわけではない．さらに，事前作業として Aim を収集することは作業者の負担となる．そこで，本研究では機械学習アルゴリズムの中でも教師なし学習の 1 つであるクラスタリングを予測に用いることで，作業の計測と同時に Aim の分類を行い，後作業として Aim のラベル付けをすることで事前のデータ計測や Aim の収集を省略する手法を提案する．

### 3.5 クラスタリング

クラスタリングとは，ある 1 つの集合を共通の特徴を持つ複数の部分集合に分割する手法である．クラスタリング手法では，クラスタリングに用いるパラメータを距離として扱う．本研究では Ward 法 [8] と呼ばれるクラスタ間の距離を定義する距離関数を使う．Ward 法は計算量が多いが分類感度が高い．以下のような手順でクラスタリングを行う．

1. すべての Task が別々のクラスタである状態から始める．
2. 今あるクラスタの中で，最も距離の近い 2 つのクラスタを選んで 1 つのクラスタに合体する．
3. 目標のクラスタ数になるまで続ける．

この手順の中でクラスタ同士の距離の決め方によって結果が変わる．その距離の決め方の手法の一つが Ward 法である．集合 P と Q があるとき，Ward 法では P と Q の距離  $d(P,Q)$  を以下のように定義する．

$$d(P,Q) = E(P \cup Q) - E(P) - E(Q) \quad (3)$$

ただし， $E(A)$  は A のすべての点から A の質量中心までの距離の二乗の総和である．式 (2) で定義された距離  $d(P,Q)$  を使って距離の近い 2 つのクラスタを一つに合体する．クラスタリングは機会学習アルゴリズムで，教師なし学習の 1 つである．クラスタリングでは事前学習なしに Task の分類ができるので作業の計測と同時に分類が行える．また，3.4 でも述べたように計測された Task 列には種類，順序，

作業量（クリック数や打鍵数）には Aim を表す特徴があり，クラスタリングはパラメータを特徴として分類する手法であるので Task の Aim ごとの分類に適している．本手法ではクラスタリングを予測に用いて事前のデータ計測や前作業としての Aim の収集の軽減を図る．

## 4 実験

### 4.1 提案手法

提案手法はTaskPitで計測されたTaskのデータをクラスタリングし，その集合にラベル付けをする．

#### 4.1.1 パラメータ

本研究では3.4節で説明した，8種類のパラメータを使用する．3.3節で説明したように，一連のTaskは1つのAimを達成するために行われる．さらに，大橋らはAimごとに行われるTaskの順序が異なるという仮説を立てて順序を特徴とするようなパラメータを3つ用意し，Aimの推定に使用した．

しかし，Aimの分類に使用する特徴がすべて等しくクラスタリングの距離の算出に有効ではない．例えば，開発作業において右クリック数などのパラメータは一般的に左クリック数や打鍵数よりも少なく，Aimの分類の際にノイズとなってしまふ可能性がある．そこで，本研究ではクラスタリングに用いるパラメータの組み合わせを変化させ，それぞれの結果から，どのパラメータがAimの分類に有効なパラメータであるかを明らかにする．

#### 4.1.2 処理手順

クラスタリングでは各パラメータの数値を距離の計算に使用するためTask名などの文字列データが推定に使えない．そこで文字列は以下の項目1のようにダミー変数を使って推定に使用している．以下の項目1では例えば”Task Name”がブラウザである場合には”Task Name. ブラウザ”が1となりそれ以外のダミー変数が0となる．

項目1

Task Name,Task Name. ブラウザ,Task Name. ファイル操作, Task Name. テキスト閲覧			
ブラウザ,	1,	0,	0
ブラウザ,	1,	0,	0
ファイル操作,	0,	1,	0
テキスト閲覧,	0,	0,	1
ファイル操作,	0,	1,	0
テキスト閲覧,	0,	0,	1

また，各パラメータの前後情報は以下の項目2のような形でAimの分類に使用している．以下の項目2では一行を1Taskとして前後10Task以内で左クリック数が一番高いTask名をダミー変数で表している．例えば，以下の項目2では6つのTaskを示していて，その中で一番左クリック数が多いTask名は”ブラウザ”であるので”T.1stLClick. ブラウザ”が1となっていてそれ以外が0となっている．

項目2

Task Name,	LClick,	RClick,	Keystroke,	T.1stLClick.	ブラウザ,	T.1stLClick.	ファイル操作	T.1stLClick.	テキスト閲覧
ブラウザ,	35,	3,	153,	1,		0,		0	
ブラウザ,	12,	0,	0,	1,		0,		0	
ファイル操作,	1,	0,	0,	1,		0,		0	
テキスト閲覧,	9,	0,	4,	1,		0,		0	
ファイル操作,	1,	0,	0,	1,		0,		0	
テキスト閲覧,	1,	1,	0,	1,		0,		0	

## 4.2 対象データ

実験には大橋らが実験で使用したのと同じデータを用いる。使用するデータは奈良工業高等専門学校5年生の学生5人のデータである。すべての被験者は研究室に配属されており、講義の課題やレポート、研究などを配属先の研究室にある割り当てられたPCで行っている。実験では被験者らが普段研究室で使用しているPCにTaskPitをインストールし、月曜日から金曜日までの5日間の被験者らのTaskおよびAimを計測した。被験者がどのAimのためにTaskを行っているのかを計測するために随時Aimを入力できる機能がある。表1に各被験者らが記録したTask数、Aim数を示す。

表1 被験者のTaskとAimの数

	被験者A	被験者B	被験者C	被験者D	被験者E
Task数	1130	637	1998	1550	1667
Aim数	8	5	5	6	5

本稿では、これらのTaskでクラスタリングを行い、各被験者のデータに対してAim数と同じクラスタ数に分ける。

## 4.3 評価

実験ではパラメータの組み合わせを変化させたときのクラスタリング結果がどのように変化するかを分析する。表2に本実験のクラスタリング結果の例を示す。表2の”設定された作業目的”は被験者らが設定した作業目的（すなわち正解）を表しており、各列はクラスタリング結果による分類先を表している。それぞれのセル内の数字は設定された作業目的の作業が各クラスタに分類された数を表している。例えば表2の1433という数字は、クラスタ1に息抜きのTaskが1433個分類されているということである。

表2 出力例

		クラスタ			
		1	2	3	4
設定された作業目的	卒論・予稿	122	375	0	87
	息抜き	1433	41	35	54
	信号処理	25	131	3	6
	ソフトウェア工学	39	0	0	35
エントロピー		2.197	1.427	0.683	2.197

ここで、各クラスタについて、以下の表3のように1つのクラスタ内にある一つの Aim が100%, それ以外の Aim が0%とできれば正しく分類されているといえる。

表3 理想的な出力

		クラスタ			
		1	2	3	4
設定された作業目的	卒論・予稿	584	0	0	0
	息抜き	0	1563	0	0
	信号処理	0	0	165	0
	ソフトウェア工学	0	0	0	74

例えば、表3ではクラスタ1が卒論・予稿の群、クラスタ2が息抜きの群、クラスタ3が信号処理の群、クラスタ4がソフトウェア工学の群となっていて正しく分類できている。このように正しく分類できているかを判断するためには各クラスタのばらつき具合を見ればよい。そこで各クラスタについて、異なる Aim を含んでいる数を元にエントロピーを求め、クラスタの乱雑さを見る。確率変数  $X$  に対して、 $X$  のエントロピー  $H(X)$  は

$$H(X) = -\sum_i P_i \log P_i \quad (\text{ここで } P_i \text{ は } X=i \text{ となる確率}) \quad (4)$$

と定義される。表の一番下の行は出力例における各クラスタのエントロピーを表している。エントロピーは0に近いほどそのクラスタが純粋であることを表しており、表2の場合クラスタ3が分類の精度が高く、クラスタ1とクラスタ4が分類の精度が低いといえる。

実験では各被験者において表2のような結果がパラメータの全組み合わせ分、計7種類出力される。本稿では各分類結果に以下のラベルを付けて表す。

1. パラメータをすべて使用した結果:全
2. 左クリックと打鍵数のみを使用した結果:左打

3. 左クリックと右クリックのみを使用した結果：左右
4. 左クリックのみを使用した結果：左
5. 右クリックと打鍵数を使用した結果：右打
6. 右クリックのみを使用した結果：右
7. 打鍵数のみを使用した結果：打

各パラメータの分類結果を比較することで分類に使用したパラメータのうちどれが重要であるかを調査する。設定された作業目的は各被験者によって異なるので作業目的ごとにエントロピーを出すのではなく、クラスタごとにエントロピーをだす。上記の各分類結果の中でエントロピーの平均を求め、使用したパラメータが同じ結果の全員分の平均をとり比較する。以下の図2にパラメータの組み合わせごとのエントロピーの出し方を示す。

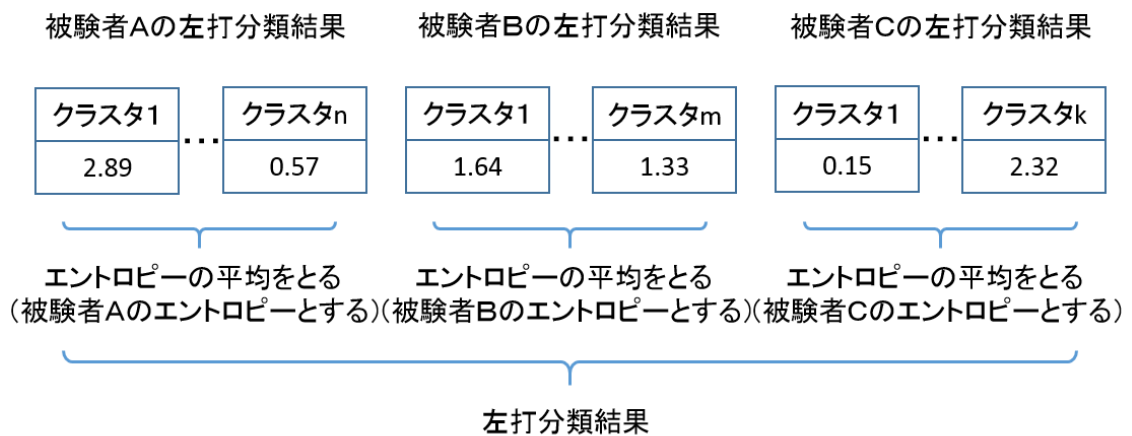


図2 エントロピーの出し方

## 5 結果と考察

### 5.1 パラメータごとの平均エントロピー

本章では各パラメータの組み合わせごとに被験者全員のエントロピーを平均したものを示すことでそれを比較し、クラスタリングにおけるAim分類ではどのパラメータが分類に有用かを示す。図3に被験者全体の各パラメータの組み合わせごとの平均エントロピーの値を示す。

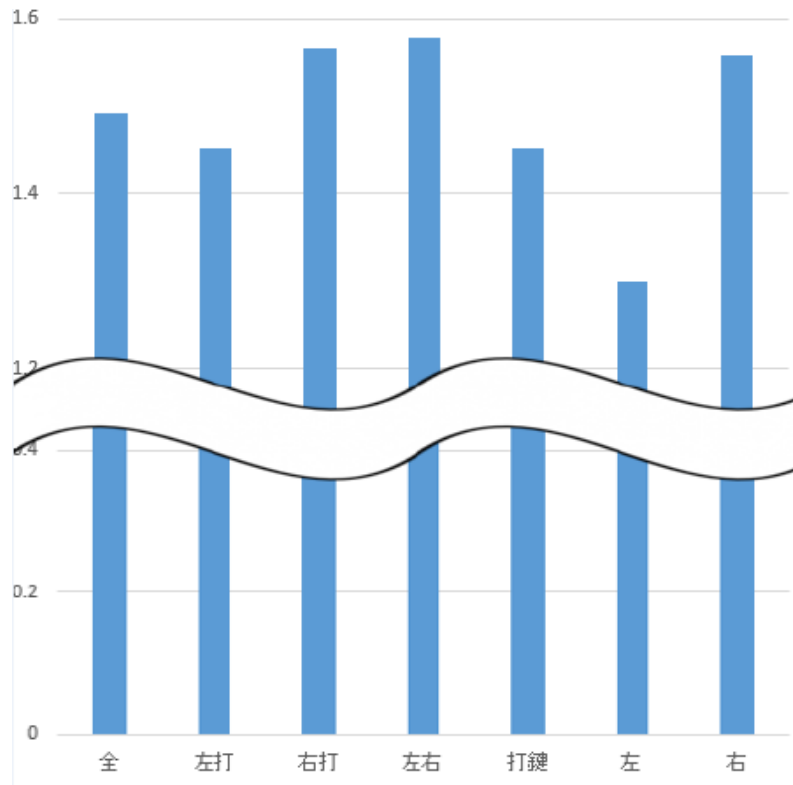


図3 エントロピーの平均

図3の縦軸はエントロピーで横軸が各パラメータの組み合わせごとの平均エントロピーを示す。図3から、全よりもエントロピーが小さいのは左打、打鍵、左であることがわかる。これら3つの結果はいずれも右クリック数を推定に使用していない。全よりもエントロピーが大きくなっているものは、右打、左右、右である。これらの結果から、右クリック数を推定に用いることでエントロピーは大きくなってしまふことがわかる。

しかし、全は右クリック数を推定に使用しているにも関わらずエントロピーは他の右クリック数を分類に使っている結果よりもエントロピーが小さい。右クリック数を分類に使うことでエントロピーが大きくなってしまふが左クリック数と打鍵数の組み合わせが分類の精度を上げていてエントロピーが下がっているのだと考えられる。左クリック数のみをつかって分類しているものは一番エントロピー

が小さく打鍵数のみを分類につかっているものは二番目にエントロピーが小さい。以上の結果より、左クリック数がクラスタリングにおけるAimごとのTask分類では左クリック数が一番重要度が高く、打鍵数が二番目に重要度が高く、右クリック数はノイズとなってしまっていることがわかる。

## 5.2 被験者ごとのエントロピー

表4に各被験者のパラメータごとの全クラスタの平均エントロピーを示す。例えば、被験者Aの”全”の1.688という値は被験者Aのデータについて、全パラメータを使って求めたクラスタが含むAimから算出される平均エントロピーを示す。

表4 分類結果

	全	左打	左右	右打	打	左	右
被験者A	1.688	1.725	1.729	1.895	1.746	1.288	1.706
被験者B	1.478	1.476	1.570	1.267	1.399	1.399	1.531
被験者C	1.315	1.199	1.339	1.458	1.168	1.175	1.353
被験者D	1.564	1.491	1.523	1.714	1.499	1.510	1.681
被験者E	1.414	1.375	1.676	1.560	1.452	1.123	1.522
平均	1.492	1.453	1.567	1.579	1.453	1.299	1.559

表4を基に被験者ごとのエントロピーに注目する。各被験者についてエントロピーが一番高いパラメータの組み合わせとエントロピーが一番低いパラメータの組み合わせについて表5に示す。

表5 被験者ごとのエントロピー

	エントロピーが大きかった組み合わせ		エントロピーが小さかった組み合わせ	
被験者A	右打	1.895	左	1.288
被験者B	左右	1.570	右打	1.267
被験者C	右打	1.458	打	1.168
被験者D	右打	1.714	左打	1.491
被験者E	左右	1.676	左	1.123

上の表5では被験者全員が右クリック数を使っているパラメータの組み合わせのエントロピーが最大であった。しかし、ほとんどの被験者について右クリック数を用いていないパラメータの組み合わせのエントロピーが最小のエントロピーとなっているが、被験者Bについては右打がエントロピー最小のものとなっている。被験者Bは全と左打のエントロピーがほとんどかわらず、打、左それぞれの結果もほとんど同じである。被験者Bの最小エントロピーが右打であることについて表6をもとに考察する。



表6 被験者の作業量

	左クリック数	右クリック数	打鍵数
被験者A	7031	219	18802
被験者B	2705	76	14947
被験者C	7119	260	18660
被験者D	8534	131	29959
被験者E	8210	665	64070

表6では各被験者ごとの左クリック数, 右クリック数, 打鍵数を示す. 表6では被験者Bのみが全体的な作業量が少なく, 打鍵数が相対的に多くなっている. 打鍵数が多いことから打鍵数が作業の特徴を捉えていて大きく分類に影響していると考えられる. 表4では, 被験者Bの打鍵数を除いた左右のエントロピーは全のエントロピーよりも大きく, 打鍵数のみを用いて分類した打はエントロピーが2番目に小さい結果となっている. 計測者の作業傾向によって結果が大きく変わるのでその人によってパラメータの重みづけを変えられるシステムがあると良いと考えられる.

### 5.3 クラスタ数

本稿ではクラスタリングを用いて分類をする際にクラスタ数を被験者の Aim の数と同じ数のクラスタ数に分類している. こうすることで後作業として各クラスタにラベル付けする際に Aim とクラスタ数が1対1の関係になる. ここでクラスタ数を増やすことでラベル付けをする後作業が増えてしまうが, 分類精度の変化を見る.

先ほど右クリック数を除いてクラスタリングした方が分類精度が高くなると述べたので, パラメータから右クリック数を除いた”左打”を対象に, クラスタ数を2倍,3倍にした時の結果を分析する. 表7に”左打”を対象にクラスタ数を2倍, 3倍にした時の被験者ごとのエントロピーの平均を示す. クラスタ数を2倍にしたときのほうがクラスタ数が Aim の数と同じときよりもエントロピーが小さくなっていることがわかる. しかし, クラスタ数を3倍にしたときは2倍のときと比べてほとんど違いがない. これは, クラスタリングによって Task から Aim を分類しようとしたときある程度までの大きさなら正しく分類できる. だが, クラスタリングでは詳細な分類まではできないと考えられる.

表7 クラスタ数を変えた結果

	左打	クラスタ数2倍	クラスタ数3倍
被験者A	1.725	1.193	1.169
被験者B	1.476	1.254	1.182
被験者C	1.199	1.226	1.222
被験者D	1.491	1.140	1.145
被験者E	1.375	1.261	1.227
平均	1.453	1.211	1.186

## 5.4 考察

実験の結果から、クラスタリングを用いたTaskからのAim推定では使用しているパラメータのうち右クリック数がノイズとなってしまっている可能性が示唆された。また、クラスタ数を増やしてクラスタリングを行ったところ、全体の平均エントロピーが減少した。つまりクラスタ一つあたりの乱雑さが減少しているということである。ただしクラスタ数を2倍にしたら乱雑さが減少しているが、クラスタ数を3倍にしてもほとんど乱雑さは減少しなくなり、クラスタリングでは細かい分類はできないことがわかった。乱雑さをなくし、さらに高い分類精度が出せないと本来の目的であるPSPの向上に計測データを使用できない可能性がある。本研究での分類精度ではまだ、PSPに適用しても正しくプロセス改善ができない可能性があるからである。なのでより高い分類精度をあげる手法の考案が必要である。クラスタリングでより細かい分類を行うためには、今回の結果を考慮してパラメータに重みづけをすることが重要であると考えられる。例えば、TaskPitに計測途中でも被験者ごとにパラメータの重みづけを変える機能を実装すれば、他とは異なる傾向を示す被験者でもより高い分類精度がでると考えられる。

## 6 おわりに

本研究ではTaskPitで作業を計測する際に、それに対する作業目的を事前に登録せず、クラスタリングを用いて自動分類することを目的とした。TaskPitが計測した作業履歴から時系列情報を含めたパラメータを作成し、Ward法を用いたクラスタリングを行う手法を提案した。

実験では、計測された作業履歴を対象に提案手法を適用し、正解データを用いて各クラスタを評価した。クラスタリングに利用するパラメータのどれが重要か明らかにするために、利用するパラメータが異なる複数のクラスタリングを行い、パラメータをすべて使った場合とエントロピーがどう変化するかをみた。

実験の結果、左クリック数、打鍵数、右クリック数の順にパラメータが重要であることが分かった。右クリック数は3つのパラメータのなかで最も重要度が低く、右クリック数を使った分類は全てのパラメータを使用した場合よりもエントロピーが増加した。また、クラスタ数を2倍にまで増加させるとエントロピーは下がったが、さらにクラスタ数を3倍にまで増加させるとエントロピーはほとんど変わらなかった。このことからクラスタリングでは詳細な分類はできないことがわかった。クラスタリングでは細かい分類ができていないので、クラスタリングをつかってPSPを使っただけの開発プロセスの改善は期待できない。そこで、クラスタリングを用いて高い分類精度を出す手法の考案が必要である。

本研究の今後の展望として、クラスタリングを用いて高い精度でAimごとにTask进行分类するためには、被験者ごとに重要なパラメータが違うので被験者ごとにパラメータの重みづけをして分類する手法の考案が必要である。また、クラスタリングだけでなくTaskPitを用いた前作業を軽減しつつより高い分類精度を出す手法の発案も必要である。より多くの被験者データを用いての分析実験、実際にプロセス改善に役立てられるかの検証などがあげられる。より多くの被験者データを使うことで異なる傾向を示す被験者がいても提案手法に分類精度を上げるための有用性があるのかを確認する。また、実際にプロセス改善に役立てられるかを検証して、高い精度や低い精度の分類結果を使ってプロセス改善を試みることで、分類の精度がどの程度プロセス改善に影響するかを調べる必要がある。そうすることである程度までの分類精度までをだせばよいという結果になる可能性がある。

## 謝辞

本論文の執筆および研究をすすめるにあたり、多くの方々に協力していただきました、この場を借りてお礼を申し上げます。ありがとうございました。指導教員である上野秀剛准教授には、この1年間を通して研究に関する知識のご教示から、アドバイス、論文のチェックまで多くの面でご指導いただきました。査読教員である岩田講師には論文の内容、鋭い質問など今後の研究に対するアドバイスをいただきました。ここに深謝の意を表明させていただきます。ありがとうございました。上野研究室の先輩方には研究や論文、学生生活に関しても様々な意見やアドバイスを下さり、充実した1年をすごすことができました。ありがとうございました。

## 参考文献

- [1] T .DeMarco, “ Controlling Software Projects: Management, Measurement and Estimation, ” Yourdon Oress, New York, USA, 1982.
- [2] ”Process DashBoard,” <http://www.processdash.com/>
- [3] ”Task Coach,” <http://members.chello.nl/f.niessink/>
- [4] ”SlimTimer Time Tracking without the Timesheet,” <http://www.slimtimer.com/>
- [5] ”TaskPit,” <http://taskpit.jp.org/>
- [6] R. Oohashi, H. Uwano ”Task Purpose Prediction from Time-series History” Advanced Course of Electronics and Information Engineering, 2014
- [7] A. Monden, Y. Kamei, H. Uwano, and K. Matsumoto, “A Software Task Measurement System for Process Improvement,” In Proc. Workshop on Foundation of Software Engineering (FOSE), 2008, pp.123128 (In Japanese.)
- [8] ”Ward 法” <http://www1.tcue.ac.jp/home1/y Miyatagbt/cluster01.pdf>