論文 / 著書情報
Article / Book Information

題目(和文)	エージェントシミュレーションのログ解析に関する研究
Title(English)	
著者(和文)	田中祐史
Author(English)	Yuji Tanaka
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第10990号, 授与年月日:2018年9月20日, 学位の種別:課程博士, 審査員:出口 弘,山村 雅幸,三宅 美博,小野 功,石井 秀明,吉川 厚,寺野 隆雄
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第10990号, Conferred date:2018/9/20, Degree Type:Course doctor, Examiner:,,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

エージェントシミュレーションの
ログ解析に関する研究

総合理工学研究科

知能システム科学専攻

田中祐史

概要

本研究の目的は「エージェントシミュレーションでの特定の入力パラメータ下で起こる異なるプロセスについての規則性の存在とその抽出手法を示す.」ことである．エージェントシミュレーションの主要な目的のひとつは，モデルが対象としている現象について，発生プロセスの理解を深めることである．しかし，エージェントシミュレーションは多くの場合は確率過程であり，実行のたびに様々な異なる挙動を示すため，そこからプロセスについての規則性を見出すのは容易ではない．本研究では，抽出すべき規則性として”シミュレーションのプロセスについての類型と，その類型と関連があるエージェントの行動”（PSP-AA ： Patterns of Simulation Processes and Actions of Agents）と定義し，エージェントシミュレーションの実行ログ中における，その存在の実証と抽出手法の提案を行う．

本論文は 5 章で構成される．概要は以下の通りである．

第 1 章「序論」では，本研究の背景・目的を示し，そこから本研究の問題を示している．

第 2 章「関連研究」では，本研究のアプローチが既存の主要な分析手法と比較して，同条件下で発生するプロセスを類型化し，その単位での分析を行うという点で，メゾレベルの分析手法であり，これが本研究独自のアプローチであると主張している．比較する分析手法として，感度分析に代表される入力パラメータと出力指標との関連を分析するマクロレベルの分析と，特定のシミュレーション試行の個別のログを詳細に分析するミクロレベルの分析を挙げている．独自性の根拠として，マクロレベルの分析ではプロセスの分析が難しいこと，ミクロレベルの分析では経験的な知見しか得られないことを挙げている．

第 3 章「提案手法」では PSP-AA を抽出する提案手法を説明し，さらに古典的なエージェントモデルである分居モデルへの適用事例にて規則性が抽出できることを確認している．そして本研究の提案手法を要約し，“モデルが扱っている現象の発生プロセス”のログを時系列クラスタリングにより類型化し，その類型ごとの“エージェントの行動”のログについての特徴を決定木分析によって取り出すこととしている．さらに，“モデルが扱っている現象を示すプロセス”，“エージェントの行動”について，エージェントシミュレーションのどの要素についての観測をログとして扱うかを，エージェントシミュレーションの標準化手法である ODD プロトコルを用いて同定する方法を提案している．また，“モデルが扱っている現象を示すプロセス”を時系列クラスタリングの際に，ログ間の距離をどう定義するかという問題に対して，ログのデータ構造に応じた妥当な時系列間の距離関数を示している．上記，提案手法を分居モデルへ適用し，特定ステップの特定セルのエージェントの行動と収束パターンが異なる 2 つの類型との関連を示すことで，エージェントシミュレーションのログ中に PSP-AA が存在することを示している．

第 4 章「提案手法の適用」では本研究で提案する抽出手法を 2 つの既存のエージェントシミュレーションに適用することで手法によって得られる規則性の妥当性・新規性を確認

している．得られた規則性について，既存のエージェントシミュレーションを用いた既存研究での結果と比較し，整合的であることをもって妥当であることを，既存研究では得られていない規則性が得られたことをもって新規性があることを主張している．1 つ目の適用例である改善・逸脱モデルでは既存研究の主要な結論を，本研究で得られた規則性からも示せたことに加えて，既存研究では論じられていない改善・逸脱のプロセスについての新たな類型を得ている．2 つ目の適用例である連鎖破綻モデルでは，既存研究ではモデルにおいて連鎖破綻が発生することを示すにとどまっていたのに対して，本研究では連鎖破綻のプロセスについての 3 つの類型を示している．

第 5 章「結論と展望」では本研究の貢献と今後の課題について説明している．提案手法の限界，課題について論じるとともに提案手法で抽出される規則性の応用可能性について論じている．

本論文では，従来から重要性は指摘されていたが解決方法が十分に示されていないエージェントシミュレーションにおけるプロセスの分析方法について，PSP-AA という論文中で定義された規則性，つまりシミュレーションの結果に大きな影響を与える特定のエージェントの行動が存在するということを，異なるシミュレーションモデルにおいて示し，同一の手法で抽出可能であることを分居モデル，改善逸脱モデル，連鎖破綻モデルという複数のエージェントシミュレーションで実証している．これはエージェントシミュレーションにおけるプロセスの分析を改善できる可能性を示している．

目次

第１章　序論

1.1 はじめに

本章では，研究の背景と目的および論文の構成を述べる．エージェントシミュレーションはモデルが対象としている現象についての，プロセスについての理解を深めることが主要な目的である．しかし，エージェントシミュレーションは多くの場合に確率過程であり，実行のたびに様々な異なる挙動を示す．ゆえに，その挙動からプロセスについての規則性を見出すのは容易ではない．本研究では，エージェントシミュレーションの多数回試行の実行ログに，プロセスについての規則性が含まれることを示すとともに，その抽出手順を示す．

本章では，本研究の目的が「本研究では特定の入力パラメータ下で起こる異なるプロセスについての規則性を抽出するための手法の提案を行う.」であることを説明する．そこから，本研究でのアプローチである「プロセスを類型化し，類型ごとの特徴を抽出する」が有望であることを説明し，本研究で解決すべき問題を導出する．具体的に抽出する規則性として，”シミュレーションのプロセスについての類型と，その類型と関連があるエージェントの行動”を示しPSP-AA(Patterns of Simulation Processes and Actions of Agents.)と定義する．

1.2 研究の目的・背景

本研究の目的は,「エージェントシミュレーションでの特定の入力パラメータ下で起こる異なるプロセスについての規則性の存在とその抽出手法を示す.」ことである．エージェントシミュレーションはエージェントと呼ばれる複数の主体をモデル化し，それらの相互作用によって引き起こされる特定の現象の分析を行う手法である．現実世界では実験することが難しい対象や，解析が困難な対象に対して有用である．適用分野は環境(Murray-Rust et al. 2014)，サービス(Rajapakse and Terano 2013)，交通(Bazzan et al. 2014)，組織(Kobayashi et al. 2013)など，多岐にわたる．

エージェントシミュレーションのモデルには確率的な挙動が含まれることが多く，初期状態が同じであったとしても，全く異なる振る舞いを示すことが多い．この振る舞いの違いことがエージェントシミュレーションの利点であり，ある初期状態から，起こりえる多様な振る舞いを示すことができる．さらに，エージェントシミュレーションではどのようなものが起こりえるか，ということに加えて，その振る舞いはどのエージェントのどのような行動によって引き起こされたかというメカニズムに迫ることができる．しかし，そもそも解析的に扱うのが難しい，複数主体の相互作用という現象の複雑さゆえ，メカニズムを明らかにする手法は未だ確立しているとは言えない．

エージェントシミュレーションは扱っている現象について，その発生プロセスについての理解を深めることが目的である．例えば，Axelrod はエージェントシミュレーションの目的につ

いて次のように述べている．*“エージェントベース・モデリングはシミュレーションの形を採用するとはいえ，特定の実験的な応用例を正確に描いて見せるのが目的ではない．それよりも，さまざまな応用例に表れる基本的なプロセスについての理解を深めるのが目的である．”*(Axelrod 1997)．つまり先述の通り，モデルが扱っている現象についてどのようなことが起こりえるか．またそれはどのようなエージェントのどのような行動が引き起こしたのかということを明らかにすることが目的と言える．しかし，エージェントシミュレーションのプロセスについての規則性を分析するという課題は，重要な課題であるにもかかわらず，近年においても解決策が示されていない課題である．

1.2.1 抽出する規則性について

本研究ではシミュレーションのプロセスについての類型と，その類型と関連があるエージェントの行動（Patterns of Simulation Processes and Actions of Agents.)」（以後，PSP-AA）を抽出する(図 1 - 1)．これは，シミュレーション結果に大きく影響を与える個別のエージェントの行動を特定できるという点で有用である．さらにこの規則性は既存研究で扱われていない，つまり存在が示されていない規則性である．

抽出する規則性：
シミュレーションのプロセスについての類型と，その類型と関連があるエージェントの行動
(Patterns of Simulation Processes and Actions of Agents.)」（以後，PSP-AA）

PSP-AAはどう有用か：
エージェントの行動の違いによって，どのようなシミュレーション結果が起こりえるかを俯瞰することが可能となる．つまり，シミュレーション結果に大きく影響を与える個別のエージェントの行動を特定することが可能となる．

PSP-AAの課題は何か：既存のアプローチでは得ることが困難であり，その存在は示されていない．

図 1-1 抽出する規則性について

出口(2013)はエージェントシミュレーションの意義について次のように述べている”*エージェントベースモデリングの多くが数理的な定式化とは対応させることができても，数理的な解析にはあまりに複雑なモデルとなっている．それゆえパラメータの影響や，さまざまな政策オプションの評価のために，相空間の全体像を見ようとするにはシミュレーションによる方法が最も有効となる．力学系で言えば，相空間の全体図を描き，その分岐構造を理解することに対応する．これはエージェントベースのモデリングにおけるシミュレーションの最も重要な役割の一つとなり，可能なシナリオの総体を俯瞰することとなる．*”本研究でも同様の立場をとる．特に，相空間の全体像を分岐させる，つまり，シミュレーション結果に影響を与える要素とし

てシミュレーションプロセスにおけるエージェントの行動に着目する．それにより，エージェントの行動の違いによって，どのようなシミュレーション結果が起こりえるかを俯瞰することが可能となる．つまり，シミュレーション結果に大きく影響を与える個別のエージェントの行動を特定することが可能となるということである．エージェントシミュレーション現実では得ることが不可能な反事実のログが得ることができるためである(Marshall and Galea 2014)．

しかし，エージェントの行動とシミュレーション結果との規則性は存在が示されていない．多くの場合，シミュレーション結果の俯瞰は，入力パラメータが要因として感度分析が行われる．しかし，エージェントの行動の場合エージェントは複数存在するうえ，ステップごとに行動が行われる．それゆえ，感度分析で要因を特定するとなると組み合わせ爆発により，入力パラメータの探索空間より膨大になる場合が多いと考えらえれ，実現が困難である．それゆえ，シミュレーションの実行ログから規則性を抽出することが有望であるが，その手法は示されていない．つまり，本研究の目的は，シミュレーション結果に大きく影響を与える個別のエージェントの行動を特定できる規則性の存在を示すことであるといえる．

1.2.2 エージェントシミュレーションのログ

本節では，エージェントシミュレーションのプロセスについてのログがどのような構造として生成されるのかについて論じる．エージェントシミュレーションはエージェントの行動・相互作用とそれによって創発する現象が互いに影響を与えながら進行していく．これをシミュレーションのログとしてとらえたとき，エージェントの行動と，創発現象という 2 つの要素の時間軸に沿ったデータがログであると考えることができる．さらにエージェントシミュレーションでは通常，シミュレーションを多数回試行してその結果を分析する．そのためログも試行回数分生成され，実行の都度その挙動は異なる．それゆえ，その分析は難しい．

そもそもエージェントシミュレーションは”*エージェント間のミクロレベルのインタラクションで創発するマクロな現象，ならびにそれがトップダウンにエージェントに影響を与えるというミクロマクロ・リンクの現象*”(寺野 2012, 2015)をあつかっている．エージェントシミュレーションにおいて自律的に意思決定を行う複数のエージェントと，それら主体間の相互作用によって創発する現象が存在する．

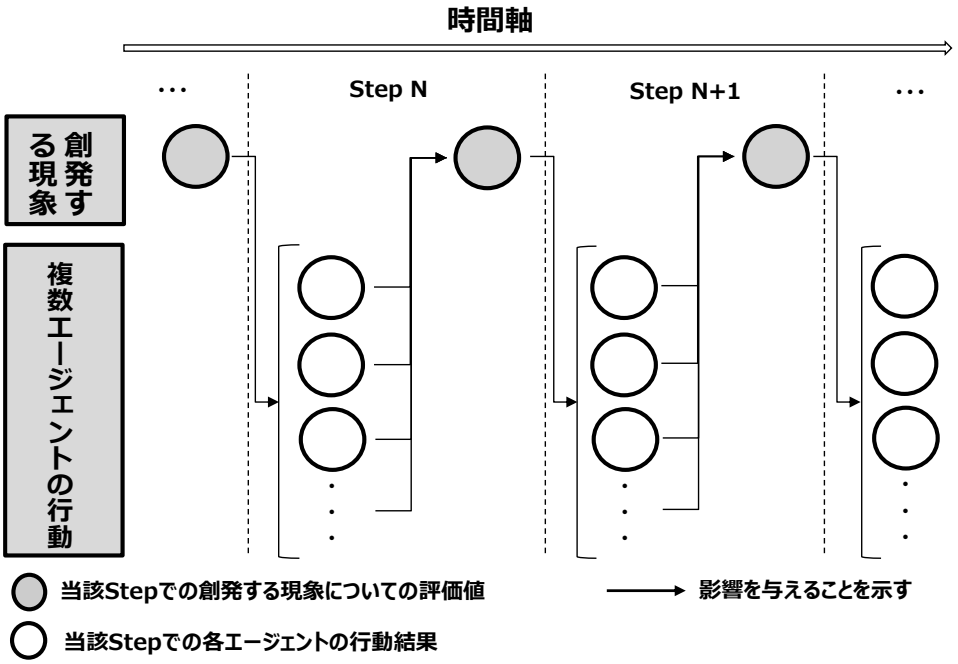


図 1-2 エージェント間の相互作用と創発する現象間の関係（ミクロマクロ・リンク）を時間軸に展開.

これらエージェントの行動と，創発現象は時間軸に沿って相互に影響を与えながら変化していっているということだと考えることができる(図 1-2)．さらに，これらは一般に“Step”や“期”といった単位で，シミュレーション上の時間軸に沿って生成される．図 1-2 はその構造を図示したものである．エージェントシミュレーションの分析にはログの分析が行われる．シミュレーションの実行において，主体間の相互作用の結果生成される全体の振る舞いや，個々のエージェントの行動の挙動がログとして生成される．このログが複数回試行の場合，同じ初期状態であっても異なるログを示す．試行とは同じ初期状態で乱数の種のみ変化させシミュレーションの 1 回の実行を指している．(Lee et al. 2015)はエージェントシミュレーションが生成する時系列が”出力データがモデルの単一コンポーネントによって生成されることはまれである”と述べ，さらにエージェントの意思決定，行動，相互作用といったミクロな要素についてのリストのほか，”マクロな要素として，ただの集計値というより，モデルが示す創発についての特徴のリスト”が生成されると述べている．これらのことから，ミクロな要素であるエージェントの行動と，マクロな要素である創発する現象は時間軸に沿って図 1-2 のように，相互に影響を与えながら時系列のログとして出力されていくと考えることができる．

一方，図 1-3 はエージェントシミュレーションにおける多数回試行時のログの構造を示している．縦軸はシミュレーション上の時間軸であり，横軸に広がる平面空間はそのシミュレーションの時点での，ログの値の取り得る空間を示している．ゆえに，1 回の試行のログは各時点での平面のいずれかの値を取り，それを結んだものが 1 試行のログとなる．図 1-3 では各ログを赤線で示しており，複数の異なる試行ログが示されている．図 1-3 で示している通り，エージェントシミュレーションでは振る舞いを示すログが生成される．必ずしも試行ごとに同一の振る舞いを示すとは限らないため，メカニズムの分析が困難である．ここで，このエージェントシミュレーションから出力されるプロセスについてのログが，シミュレーションを多数回試行した場合どのような出力になるかを考える．エージェントシミュレーションは確率過程である(Lee et al. 2015)．それゆえ，図のように，初期状態が同じであったとしても異なるプロセスを経て最終状態も異なる．図が示す Step ごとの空間は，分かりやすさのため簡易的に平面空間で示しているが，実際には前項で述べたように複数の時系列で構成されているため，それらの取り得る値の組み合わせであり高次元の空間である．それゆえ，ここからプロセスについての特徴，規則性を見出すのは困難である．そのため，ログを単純に図示するのは不可能であり，視覚的に特徴を見出すことはできない．この対象の分析の困難さは(Lee et al. 2015)や(Szimanski et al. 2013)で指摘されている．

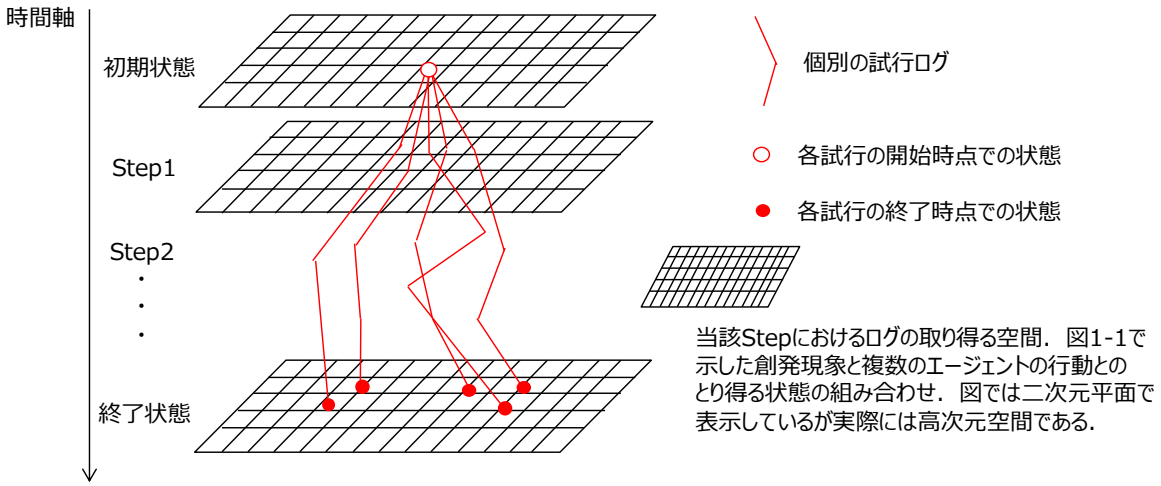


図 1-3 エージェントシミュレーションにおける複数回試行時のログ（図は 5 試行のログ）

1.3 既存のエージェントシミュレーションにおける分析の課題

前項で示したエージェントシミュレーションのプロセスにいての複雑さに対して，特定の 1 試行のログを取り出して分析(図 1-4)することが行われている．この分析では，時間的に変化していくエージェントの行動と，それによって引き起こされる現象を，詳細に追うことができる．（例えば，エージェント A が行動 XX を，エージェント B が行動 YY を行った．その結果，ZZ という現象に至った．さらに，ZZ という現象は，エージェント C の内部状態に影響を与え，・・・といったように）．しかし，ログ分析から分かることはあくまで単一のログで起こったことであり，エージェントの行動が偶然であったのか，必然であったのかはわからない．そのため，本研究が目的とする規則性を抽出することは難しい．

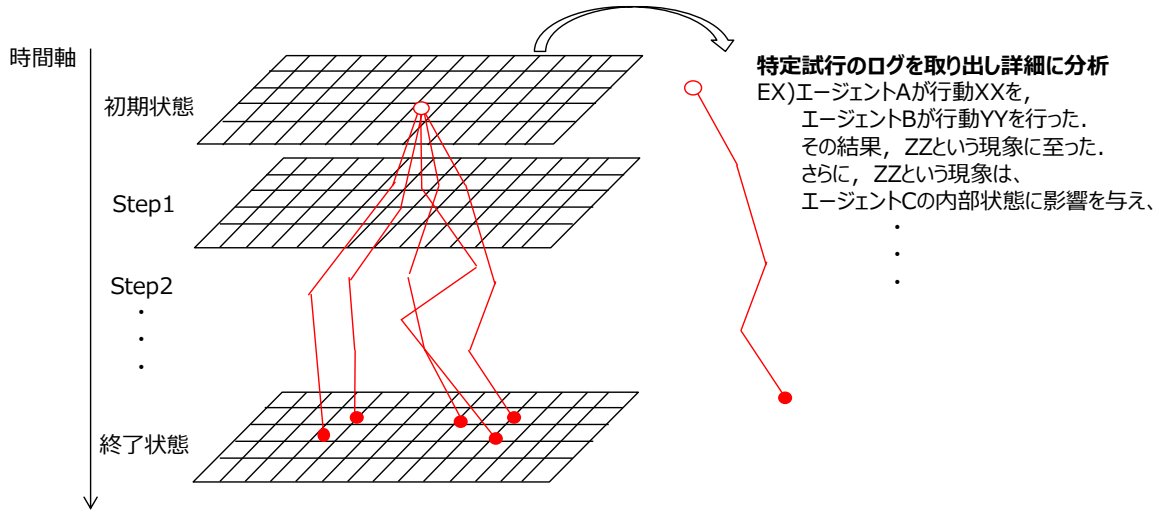


図 1-4 個別試行のログ分析

1.4 研究の問題

個別試行のログ分析は扱っている対象が一事象なので，法則が分からないことに着目し，本研究では「シミュレーションのプロセスを類型化し，その単位での法則を分析する」というアプローチを考える．それにより「シミュレーションのプロセスについての類型と，その類型と関連があるエージェントの行動（Patterns of Simulation Processes and Actions of Agents.)」（以後，PSP-AA）という規則性が得られると考えられる．

PSP-AA の定義と近いものとして，(Grimn et al. 2005)によるパターンがある．Grimn らはモデルの妥当性を高めるために，シミュレーションで観察されるパターンを用いることを提案している．Grimn らによるパターンの定義は“非ランダムな任意の構造であり，それが発生する

メカニズムも含む”とある．このパターンを現実世界の観測と異なる複数の観点で比較することでモデル構築を行うことを提案している．本研究における PSP-AA はこのパターンをより具体化したものだともいえる． “非ランダムな全体の振る舞いと，それのエージェントの意思決定（行動）による要因”をエージェントシミュレーションのログから抽出することが本研究の目的である．

冒頭で述べた通り，本研究の目的は「エージェントシミュレーションでの特定の入力パラメータ下で起こる異なるプロセスについての規則性の存在とその抽出手法を示す．」ことであり，抽出する規則性は，先に定義した PSP-AA である．そのためのアプローチとしてエージェントシミュレーションのログをプロセスについて類型化し，類型単位での特徴を分析することを試みる．

本研究で扱う問題は以下である(図 1-5)．エージェントシミュレーションのログからプロセスについての規則性「PSP-AA」を抽出する手法を提案し，複数の既存のエージェントモデルへ適用することで(1)を示す．(1)エージェントシミュレーションのログにプロセスについての規則性が存在する場合がある．さらに，得られた規則性について(2)，(3)を示すことで，得られた規則性の妥当性と新規性を示す．(2)得られた規則性が既存研究の結果と整合的である．(3)既存研究では示されていない規則性を得れる場合がある．

目的：エージェントシミュレーションでの特定の入力パラメータ下で起こる異なるプロセスについての規則性の存在とその抽出手法を示す．
抽出する規則性：
シミュレーションのプロセスについての類型と，その類型と関連があるエージェントの行動 “PSP-AA“ (Patterns of Simulation Processes and Actions of Agents.)

アプローチ：エージェントシミュレーションのログをプロセスについて類型化し，類型単位での特徴を分析する

問題：エージェントシミュレーションのログからプロセスについての規則性「PSP-AA」を抽出する手法を提案し，複数の既存のエージェントモデルへ適用することで以下を示す．
(1) エージェントシミュレーションのログにプロセスについての規則性が存在する場合がある．
→デモンストレーション，適用例１，２で示す．

さらに，得られた規則性について以下を示すことで，得られた規則性の妥当性と新規性を示す．
(2) 得られた規則性が既存研究の結果と整合的である．
(3) 既存研究では示されていない規則性を得られる場合がある．
→適用例１，２で示す．

図 1-5　本研究が取り扱う問題

本研究の構成は以下である．まず，第 2 章「関連研究」では，本研究のアプローチの独自性を既存手法と比較することで示す．第 3 章「提案手法」では，提案手法を説明し，古典的なエージェントシミュレーションである分居モデルに適用し，手法のデモンストレーションを行う．第 4 章「提案手法の適用」では，提案手法を既存の 2 つのエージェントシミュレーションに適用し，得られる規則性の妥当性と新規性を示す．第 5 章「結論と課題」では，本研究の貢献と今後の課題を説明する．

第 2 章　関連研究

2.1 はじめに

本章では，本研究のアプローチが既存の主要な分析手法比較して，メゾレベルの分析手法であるという点で，新規性があることを説明する．比較する分析手法として，感度分析に代表される入力パラメータと出力指標との関連を分析するマクロレベルの分析と，特定のシミュレーション試行の個別のログを詳細に分析するミクロレベルの分析を挙げる. 新規性の根拠として，マクロレベルの分析ではプロセスの分析が難しいこと，ミクロレベルの分析では経験的な知見しか得られないことを挙げる．

2.2 本研究のアプローチの独自性

本研究のアプローチであるエージェントのプロセスについてのログを類型化し，その類型ごとに特徴を得るという方法は，特定の入力パラメータで得られるログを類型化しているという点でメゾレベルの分析といえる．これは既存分析手法である多数の入力パラメータについての出力の分析であるマクロレベルの分析と，個別試行のログを詳細に分析するミクロレベルの分析に対比している(表 2-1).

表 2-1 他分析手法との違いについての整理.

	マクロレベルの分析： 感度分析	メゾレベルの分析： 本研究	ミクロレベルの分析： 個別試行のログ分析
ログの収集範囲	広範囲の入力パラメータ	特定の入力パラメータ	1つの試行
ログの扱い方	試行ごとの評価値と入力パラメータとの関連を分析.	ログを類型化し，類型間を比較することで類型ごとの特徴を得る.	対象とする試行内での因果を詳細に追う.

これら 3 つの分析手法は，扱うログの範囲が異なるがそれは分析手法を適用する目的が異なるためである．マクロレベルの分析においては，入力パラメータと出力指標との関連を分析する．これは，モデルの挙動を広域的に観測することが目的である．また，ミクロレベルの分析では何かしらの方法で選択した特定の個別試行のログを詳細に分析する．これはその試行において，全体を見ているだけでは分からない細かな挙動を確認するためである．

本研究でのメゾレベルの分析は, 特定の入力パラメータにおいて生成されるログを対象とし, ログをプロセスについて類型化し, 類型間を比較することでそれぞれの類型の特徴を得る(図 2-1)．これによってミクロレベルの分析でも，マクロレベルの分析でも得ることが難しかったプロセスについての規則性である PSP-AA を得ることを試みる．

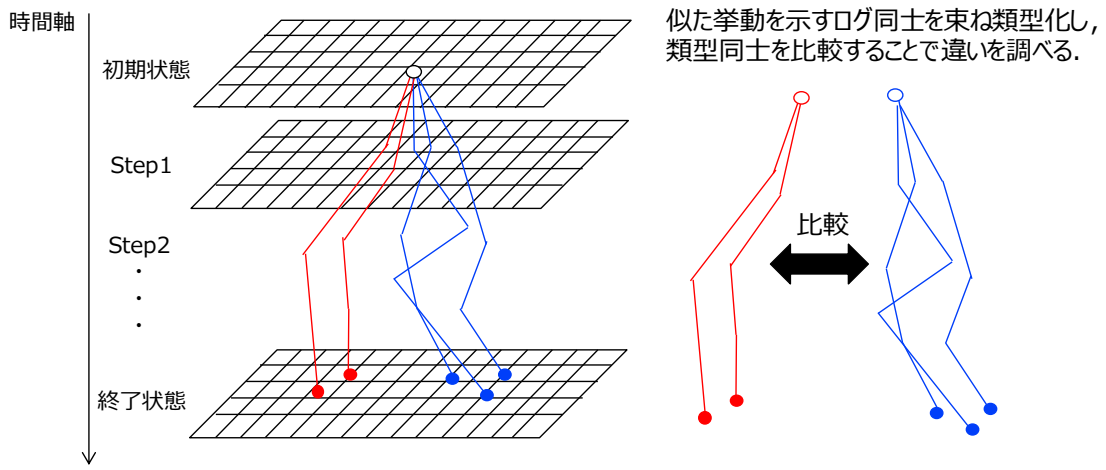


図 2-1 プロセスを類型化し，類型間を比較することで類型ごとの特徴を得る．

2.2.1 マクロレベルの分析手法：感度分析

マクロレベルの分析は，入力パラメータと出力指標との関連を分析する．これにより，モデルの挙動を入力パラメータ組み合わせ空間が広いという意味で広域的にとらえることができる．しかし，シミュレーションのプロセスについてはブラックボックスとして扱ってしまっている(Topping et al. 2010)ためプロセスについての分析が難しい(図 2-2)．

マクロレベルの分析では入力パラメータを変更していき，評価指標がどのように変化するかを観測するが，入力パラメータは通常複数あり，多段階または実数値であるため探索空間が状態爆発を起こす．最も一般的な方法は手動で探索範囲を決める試行錯誤法(Lee et al. 2015)である．このアプローチは，分析者がモデルの挙動や出力について十分に理解している場合，時間的に効率的であり得る．モデルの作成者やモデルが扱う対象のドメインの知識が豊富な分析者は，現象に関連する仮説を生成して，それに基づいた入力パラメータを選定し，仮説の正しさを検証するというプロセスをとることが可能である．ゆえに探索するパラメータの範囲を絞ることができ，結果的に必要なシミュレーション実行の回数を抑えることが可能である．しかし，このアプローチでは，人間のバイアスや疲労の影響を受け易いため，モデルのパラメータ空間の大部分が無視される恐れがある．ゆえに，より体系的で自動化されたバイアスのないアプローチが必要である．

入力パラメータはその間隔の取り方や，範囲，パラメータ間の交互作用などにより，容易に組み合わせ爆発を起こす．それゆえ，効率的な方法が提案されている(Lorscheid, Heine and Meyer 2012) (Happe 2005). (松島ら 2015)(Yamashita et al. 2014)は実験計画法の応用に着目し，少ない実験数とその結果から，相転移の様に急激に出力が変化するような入力パラメータ

の組み合わせを探索する手法を提案し，例題として避難シミュレーションの実験を通し，提案手法の有効性を検証している．具体的に，避難完了時間と，入力パラメータとの関連を分析している．ただし，実験計画法を用いる場合は入力パラメータごとの水準をあらかじめ定めなくてはならないという課題がある．エージェントシミュレーションにおいて出力への影響の大きさが異なる範囲の入力パラメータの水準をあらかじめ想定することは必ずしも容易ではない．そのため入力パラメータの値をランダムに決定しながら，実験計画を行う手法も提案されている(Thiele, Winfried and Grimm 2014) (Sobol 1993).

このようにマクロレベルの分析，つまりシミュレーションのプロセスをブラックボックスとして扱い，入力パラメータの組み合わせを変化させながら，出力を観測することで，入力パラメータと出力の関係を分析し，モデルの挙動を確認する分析においての主眼はどう効率的に入力パラメータの組み合わせを探索するかであり，シミュレーションのプロセスをどのようにして詳細に把握しようかといったことはそもそも主眼とされていない．

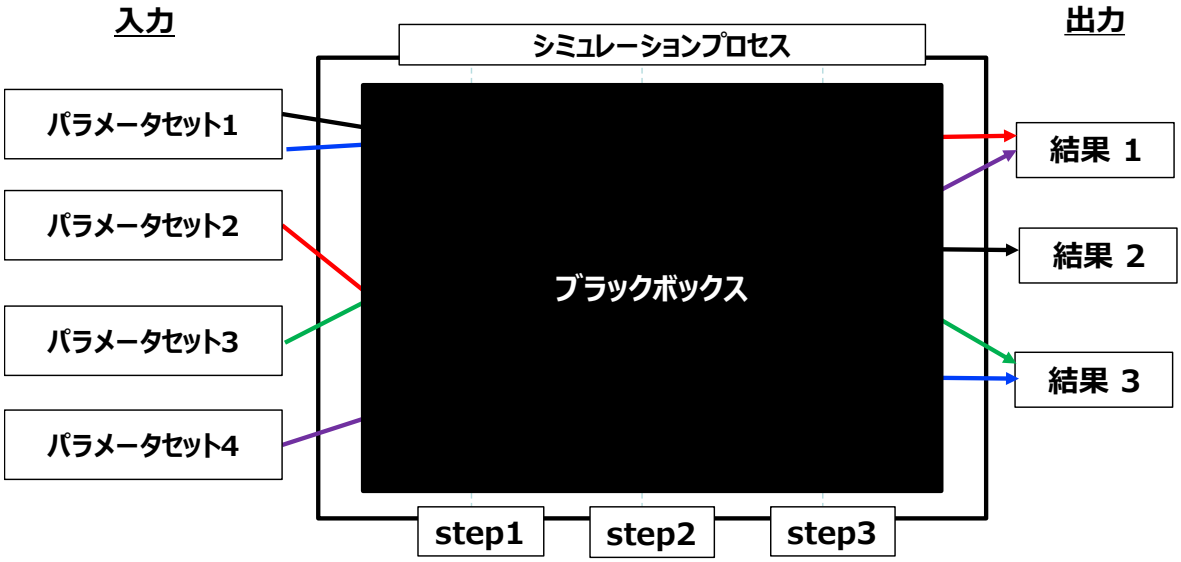


図 2-2 マクロレベルの分析手法はモデルをブラックボックスとして扱う.

2.2.2 ミクロレベルの分析：個別試行のログ分析

ミクロレベルの分析では，マクロレベルの分析で行うことが難しいシミュレーションのプロセスの分析を行うことが可能である．この分析では，何らかの方法で選択した個別試行のログを対象として，詳細な分析を行う．対象の試行で起こったことの因果の連鎖を，ログを見ることで詳細に追っていくことが可能である．具体的には時間軸に沿ったエージェントの行動の連鎖であるためプロセスについての理解が容易である．しかし，あくまで分析対象のログは 1 つであり，その試行の中で何があったかは分かるものの，起こったことが偶然に起こったのか，それとも偶然ではなく特定の要因によって引き起こされたのかといった，規則性についての分析ができない（図 2-3）．

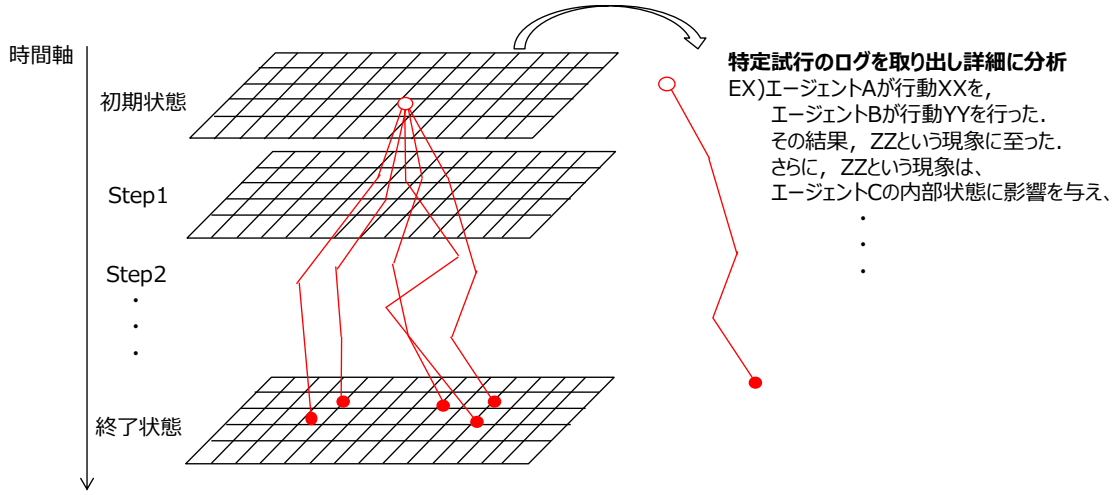


図 2-3 特定試行のログを取り出して分析するミクロレベルの分析

エージェントシミュレーションは既存の理論では説明が難しい現象を説明できる(寺野 2003,2010)．この意味において，個別試行のログ分析は，モデルが対象とする既存の理論で説明が難しい現象の具体例を示しているといえる．エージェントシミュレーションを仮想世界ととらえ，特定の試行はその仮想世界で数ある起こりえたことのうちの 1 つのケースを示していると考えることができる．

このミクロレベルの分析では，特定の事象が発生しえることを示したり，現実の事象と比較を行うことでモデルの妥当性を高めることに用いられる．(菊池ら 2016) ではモデルにおいて特定の事象が発生することを示している(図 2-4)．また，(Kobayashi et al. 2013) (Terano et al. 2013) は実在するケースと特定のログの内容を照らし合わせることでモデルの説明範囲を確認し，妥当性の向上につなげている．

また，(和泉ら 2013, 2017)は分析対象とするログの選択手法について提案を行っている．モデルの実行を途中で一時停止させ，そこから分析者が興味深い範囲を絞り，実行を続けるとい

うことを繰り返すことで，分析者が意図的に好ましい状態を作りだし，ログを抽出することが可能となる．

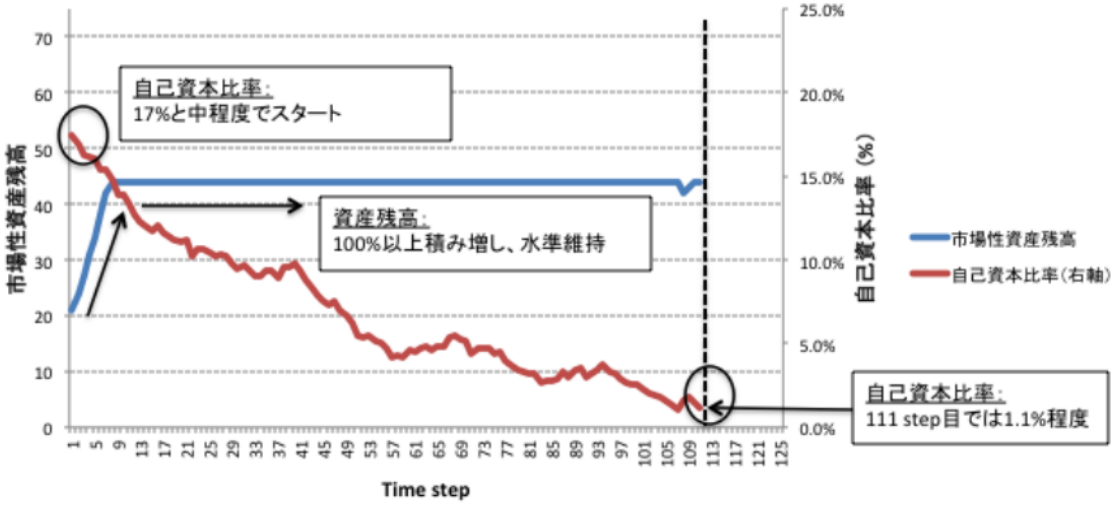


図 2-4 ログ分析を用いた分析の例(菊地ら 2016)

2.3 その他分析手法

前節で紹介した分析手法以外の，エージェントシミュレーションで典型的な分析手法として分布を用いた分析と，モデル同化の説明を行う．分布を用いた分析では，出力を分布によって可視化し，その特徴を取り出す．モデル同化は現実から得られたデータについて，そのデータと最も似ている出力をモデルが生成する条件を最適化計算などによって求める．

2.3.1 分布を用いた分析

分布を用いた分析では，多数回試行のシミュレーションの出力を分布として可視化し，そこから特徴を見出す．マクロレベルの分析では多数回試行の出力を分析手法が扱える形で集約しているが，分布を用いる場合は可視化が目的であるので，出力を集約せずに分布として扱う．分布として可視化することにより，シミュレーションの挙動が想像しやすくなり，シミュレーションのプロセスで何が起こっているかについての仮説を生成しやすくなる．しかし，多次元の出力の分布は描写が困難である点と，出力が得られた要因を明確にできない点があげられる．

分布を用いた分析（Ohori et al. 2012）（坂平，寺野 2014）(Kahl and Hansen 2015)では，出力を分布によって可視化し，その特徴を取り出している．出力は必ずしも単一の変数であるわけではなく，例えば (Ohori et al. 2012) では時系列に対してその分布を観測している(図 2-5)．図 2-5 の例ではシミュレーションの時間変化と出力の変化の関係についての傾向をとらえることが可能である．また，出力は必ずしも正規分布に従うとは限らないため，中央値，四分位点のほかに，データの分布密度も同時に確認できるヴァイオリン図(Hintze and Nelson 1998)が用いられることもある(Kahl and Hansen 2015)．

しかし，分布を用いた分析はあくまでどのような挙動が含まれるのかを確認するのが主眼であり，その要因については明らかにすることができない．さらに，可視化し，人が特徴を取り出すということが前提となっているため，特徴が多くなり，可視化が難しくなると分析が難しくなるという欠点もある．

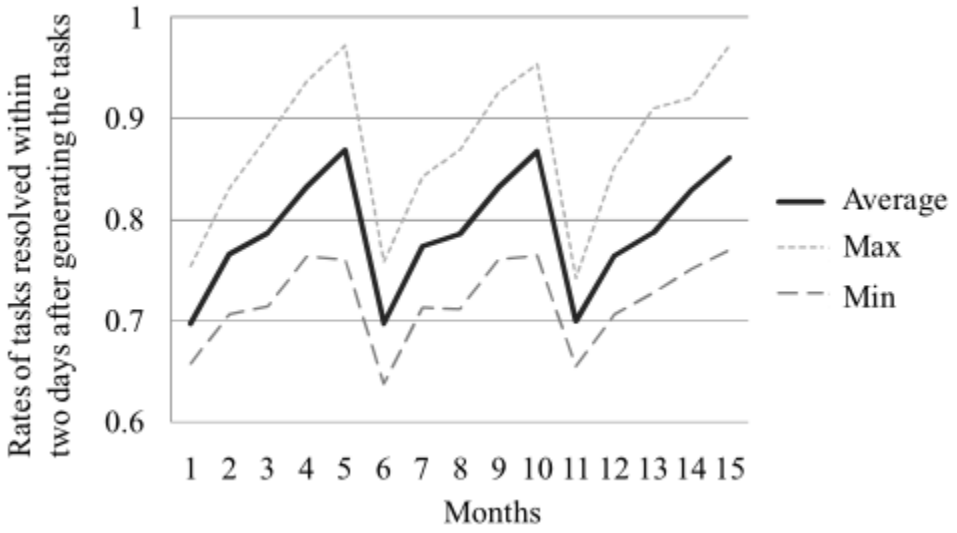


図 2-5 分布を用いた分析の例(Ohori et al. 2012)

2.3.2 モデル同化

モデル同化では，現実に近しい挙動を再現できるパラメータの探索を行う．それにより，モデルで用いるのに妥当なパラメータの範囲を同定したり，パラメータについてのある挙動の発生要因を明らかにすることができる．

モデル同化の基本的な手法としてはキャリブレーションと呼ばれ，ランダムサンプリングによりパラメータ空間を試行錯誤的に探索する（例えば，Molina et al. 2001）．分析を実施する難易度が低いため度々用いられるが，組み合わせ爆発が起こるエージェントシミュレーションのパラメータ空間においては有用であるとは限らない．

それゆえ，データと最も似ている出力をモデルが生成する条件を出力に対する評価関数を設定することで，最適化計算によってモデル同化を行うこともある．エージェントシミュレーションにおいては逆シミュレーションと呼ばれる(Yang et al. 2009) (Benoit and Hutzler 2005) (倉橋 2013)．モデル同化手法は次のような手続きで実行される(倉橋 2013)（図 2-6)．

1. 実世界を表現する多数のパラメータによるモデル設計.
2. 実際に用いられる評価関数の設定.
3. 評価関数を目的関数としてシミュレーションを実行.

4. 得られた初期パラメータの評価.
しかし，このように多数のパラメータを目的関数に対して調整することは一般に困難である．そこで逆シミュレーションでは複雑かつ多変数な関数の最適化が可能な進化計算手法や強化学習手法を採用している．

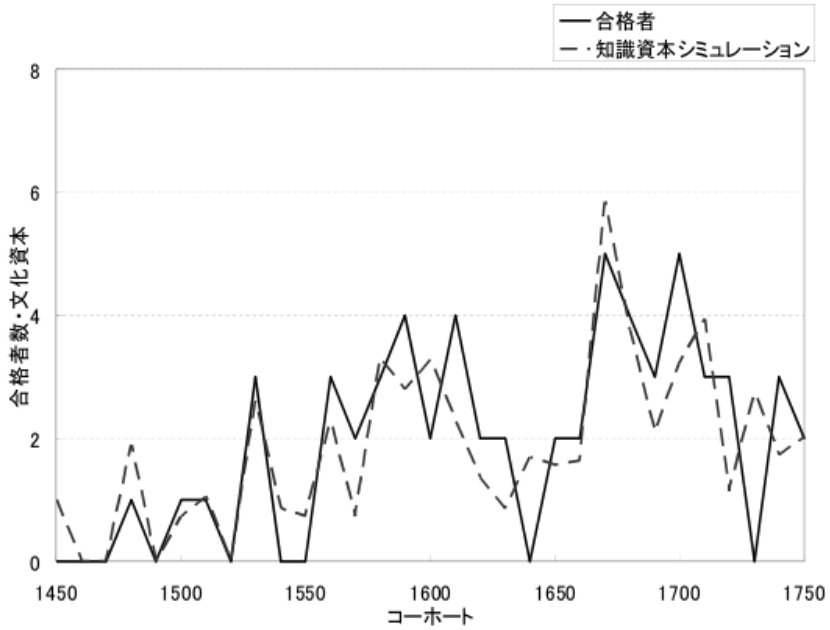


図 2-6 モデル同化を用いた分析の例(倉橋 2013) (Yang et al. 2009)

そのほかの手法として，エージェントシミュレーションの出力の尤度関数を求める手法があるが，エージェントシミュレーションは一般に複雑な確率的シミュレーションモデルであるため古典的な最尤推定を適用が困難である．そのため，近似ベイズ計算（Approximate Bayesian Computing：ABC）と呼ばれる方法が用いられる(Hartig et al. 2011) (Martínez et al. 2011) (May et al. 2013)．近似ベイズ計算では，シミュレーション結果と観測データとを，シミュレーションから計算されたいくつかの集約された情報であるいわゆる要約統計量と観測データとを用いて比較し，データの次元性を低下させる．シミュレーションされた要約統計量と観測された要約統計量との間の差が定義された閾値未満であるパラメータ値のみが保持されることにより，事後分布の近似を得る．

第 3 章　提案手法

3.1 はじめに

本章では第 1 章で定義した本研究で扱う規則性である PSP-AA を抽出する提案手法を説明する．さらに古典的なエージェントモデルである分居モデルへの適用事例にて規則性が抽出できることを確認する. 本研究の提案手法を要約すると, “モデルが扱っている現象の発生プロセス”のログを時系列クラスタリングにより類型化し，その類型ごとの“エージェントの行動”のログについての特徴を決定木分析によって取り出すことである．

ここで, “モデルが扱っている現象を示すプロセス”, “エージェントの行動”について，エージェントシミュレーションのどの要素についての観測をログとして扱うかを，エージェントシミュレーションの標準化手法である ODD プロトコルを用いて同定する．また，“モデルが扱っている現象を示すプロセス”を時系列クラスタリングの際に, ログ間の距離をどう定義するかという問題に対して，ログのデータ構造に応じた妥当な時系列間の距離関数を示す.

提案手法を分居モデルへ適用し，特定ステップの特定セルのエージェントの行動と収束パターンが異なる 2 つの類型との関連を示すことで，エージェントシミュレーションのログ中に PSP-AA が存在することを示す.

3.2 提案手法の概要

本研究のアプローチは, エージェントシミュレーションのログをプロセスについて類型化し, 類型間を比較することでそれぞれの特徴を得るということは，第 2 章で説明した．本節では，それを具体化した手法である“モデルが扱っている現象の発生プロセス”のログを時系列クラスタリングにより類型化し，その類型ごとの“エージェントの行動”のログについての特徴を決定木分析によって取り出すことを説明する．

エージェントシミュレーションのプロセスについてのログは，第 1 章において図 1-2 で創発する現象と，複数のエージェントごとに時系列で生成されることは説明した．それに従い，多数回試行のプロセスについてのログの構造は図 3-1 のようになる．創発する現象と，複数のエージェントの行動のログがステップごとに生成され，さらに同様の構造のログが試行回数分生成される．ここでモデルのどの要素が創発する現象や，複数のエージェントの行動に相当するのかという問題が発生する．提案手法では，エージェントシミュレーションの標準化手法である ODD プロトコル(Grimm et al. 2006,2010)を用いて，ログとして観測すべき要素の同定を行っている．

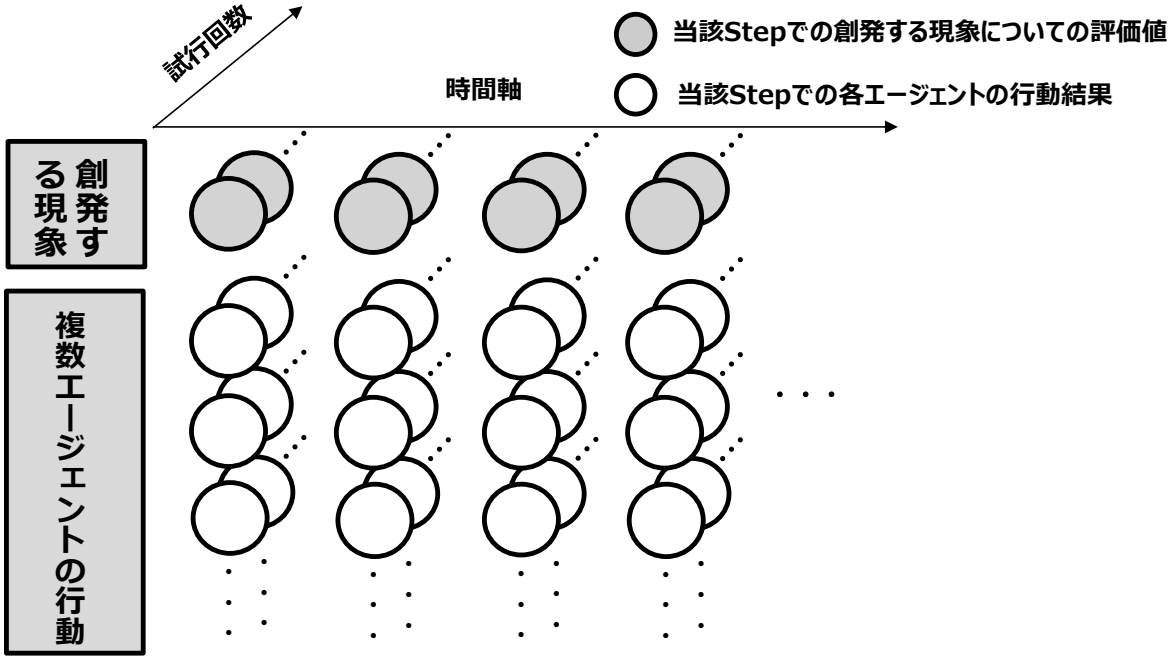


図 3-1 エージェントシミュレーションのログの構造

提案手法では，図 3-1 で示される構造のログに対して，“モデルが扱っている現象の発生プロセス”のログを時系列クラスタリングにより類型化し，さらにその類型ごとの“エージェントの行動”のログについての特徴を決定木分析によって取り出す，という 2 段階の手順によって実現される．この手順によって，ログのプロセスが類型化され，さらに類型ごとのエージェントの行動の特徴が抽出される．つまり，規則性 PSP-AA が抽出される．

ログのプロセスを類型化するに際し，主要な挙動を対象とすることが重要であり，対象とするエージェントシミュレーションが扱おうとしている創発する現象についてのログを扱う．そのため，創発する現象についてのログのみを対象とし時系列クラスタリングを行うことでプロセスについての類型化を行う．時系列クラスタリングとは，時系列データに対するクラスタリング行う手法であり，時系列のパターンを見つけることができ，分類に対する知識がなくても実行可能である(Aghabozorgi, Shirkhorshidi and Wah 2015) (Everitt et al. 2011)．時系列データは，各時点の値が 1 次元になるため高次元可能性が高い点が特徴である．それゆえ，時系列データは複雑であると言え，分析者がデータから特徴を見出すのが難しい対象である(Rani and Sikka 2012)(Liao 2005)．時系列クラスタリングを用いることで，複雑なデータである時系列データから時系列のパターンを見つけやすくする．また，ログ間の距離をどう定義するかという問題に対して，ログのデータ構造に応じた妥当な時系列間の距離関数を示している．

さらにその類型ごとの“エージェントの行動”のログについての特徴を決定木分析によって取

り出す．具体的には，クラスタリングにより得られた類型を目的変数，"エージェントの行動"を説明変数として，決定木分析を行うことで，類型ごとの"エージェントの行動"についての特徴を得る．決定木分析とは，分類手法であり最も一般的な手法の一つである(Patal and Rana 2014)．分類手法とは，あらかじめ正解（目的変数）が分かっているデータに対し，説明変数との写像を学習によりモデル化することで，説明変数だけで，目的変数を予測する手法の総称である．特に，決定木分析の特長は，学習されたモデルが，木構造で出力されるため，どのような法則を学習したかの解釈性が高く，データからの知識抽出手法としても有用性が高い点である．そのためシンプルでかつ最も成功した機械学習手法だとも言われている(Russell and Norvig 2016)．

教師なし学習であるクラスタリングと，教師あり学習である決定木分析の組み合わせは，教師データのないデータセットへの分類タスクとして用いられることが多い．例えば，(Thomassey and Fiordaliso 2006)(Yasami and Mozaffari 2010)が挙げられる．ただし，本研究での提案手法は分類が目的ではなく，規則性を得ることでデータセットに対する理解を深めることを主眼としている．

また，ログからの規則性抽出とは知識発見（knowlede dicovery in databases:KDD)ともとらえられる．知識発見のプロセスはデータ選択，前処理・データ加工，データマイニング，解釈・評価という順に行われる(Fayyad 1996)．クラスタリングが前処理・データ加工，決定木分析がデータマイニングに相当する．

3.3 提案手法の詳細

本節では，提案手法の詳細な手順について説明する．提案手法の手順についての全体像は図 3-2 である．まず，ODD プロトコルによって同定された観測項目について，特定パラメータの多数回試行を行うことで，ログを生成する．次に，生成されたログのうち“モデルが扱っている現象を示すプロセス”について階層的クラスタリング ward 法により類型化を行う．この時，階層的クラスタリングで用いる距離関数はログのデータ構造によって決定する．次に，クラスタリングによって得られた類型を目的変数，”エージェントの行動”のログを説明変数として，決定木分析を行う．これらの手順により類型と，その類型の特徴であるエージェントの行動，つまり本研究で抽出しようとしている規則性である PSP-AA が得られる．

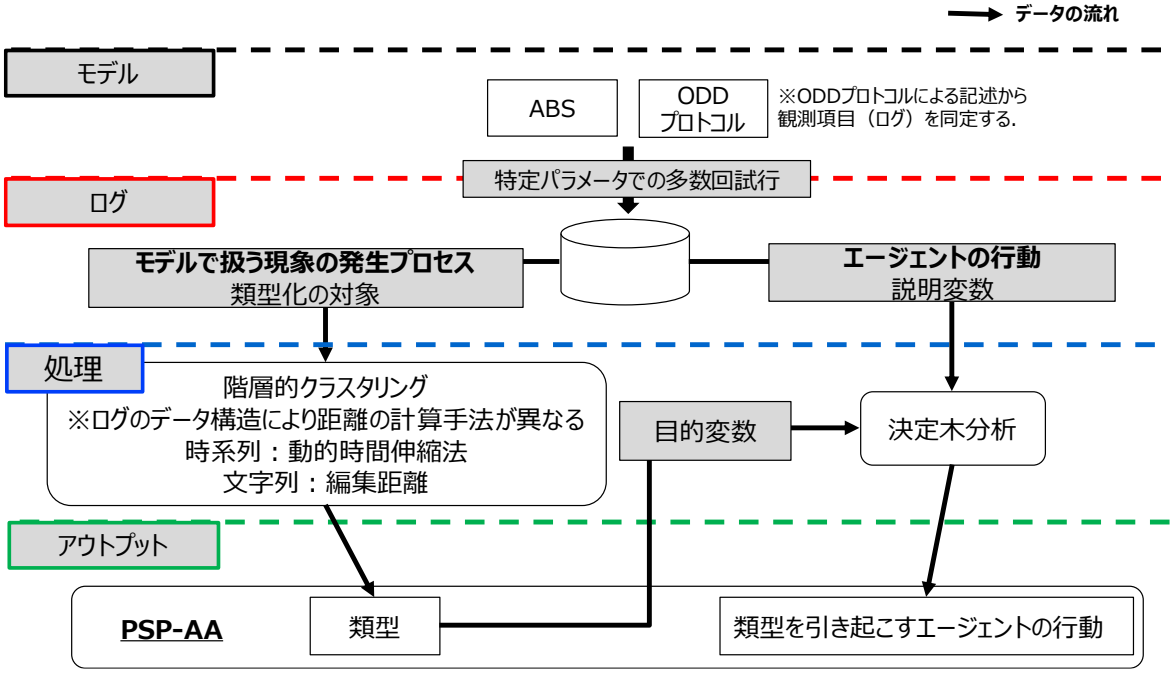


図 3-2 提案手法の全体像

ODD プロトコルの項目のうち，2 つのログは“モデルが扱っている現象を示すプロセス”は ODD プロトコルの「Entities, state variables, and scales」,「Emergence」,「Observation」から参照する．“エージェントの行動”は ODD プロトコルの「Entities, state variables, and scales」,「Process overview and scheduling」,「Observation」から参照する．

“モデルが扱っている現象を示すプロセス”は必ずしも時系列データとして得られるとは限らない．本手法では，具体的には，時系列データに対しては，動的時間伸縮法を用い，それ以外の場合はルールベールでログを文字列に変換する手法を提案し，文字列データに対してレーベンシュタイン距離を用いる．また，決定木分析のアルゴリズムとして C4.5 を用いている．

3.3.1 ODD プロトコルからの観測（ログ）の同定

ODD(Overview, Design concepts and Details)プロトコルとは，（Grimm et al. 2006,2010）によって提案された，エージェント・ベース・モデルの標準化手法である．“モデルが扱っている現象を示すプロセス”，“エージェントの行動”について，エージェントシミュレーションのどの要素についての観測をログとして扱うかを，エージェントシミュレーションの標準化手法である ODD プロトコルとの対応付けは以下のようにおこなう．

“モデルが扱っている現象を示すプロセス”は ODD プロトコルの「Entities, state variables, and scales」,「Emergence」,「Observation」から参照する．“エージェントの行動”は ODD プロトコルの「Entities, state variables, and scales」,「Process overview and scheduling」,「Observation」から参照する．

ODD プロトコルについて

ODD プロトコルはモデル構造の記述手法(Müller et al. 2014)の 1 つである．(Müller et al. 2013, 2014)はモデルの記述手法は多数存在するが，ほとんどの手法はモデルの再現が主眼となってると述べている．本研究においてモデル構造はモデルの出力の評価をどう行うかの観点を示すという目的で用いている．(Müller et al. 2014)は，この目的を”theory building”とし，その要件としてモデルの仮説，応用している理論，モデルのコンセプト，モデルの原則の記述をあげている．さらに，Müller et al. (2014)はこの記述を満たす手法として，ODD プロトコルをあげている．ODD プロトコルとは，（Grimm et al. 2006,2010）によって提案された，エージェント・ベース・モデルの標準化手法である．記述すべき要素を表 3-1 に示す．ODD プロトコルでは表 3-1 の要素を自然言語にて記述する．

提案手法において，特に重要なのはどの観測項目を用いて分析を行うかという観測の項目である．この観測項目をどう扱うかを，記述されているモデルの目的など検討する．この項目の観測項目を用いた分析は, マクロレベルの分析, ミクロレベルの分析においても通常行われる．マクロレベルの分析においては，その観測データを集約し，分析しやすくしているが，当然データの情報量は減っている．一方，ミクロレベルの分析においては，扱うデータを特定試行に絞ることでの観測データをそのまま扱い，詳細な分析を可能としている．しかし，単一のログごとにデータを扱わなくてはならない．

表 3-1 ODD プロトコルの構成(Grimm et al. 2006,2010).

	要素	説明
Overview	Purpose 目的	モデルの目的.
	Entities, state variables, and scales エンティティ，状態変数，スケール	モデルに含まれるエンティティ（Agent)と 特徴付けれらる変数や特性.
	Process overview and scheduling プロセス，スケジューリング	シミュレーションのプロセス. （Agentがどういった順番に何をやるのか）
Design concepts	Basic principles 基本原則	モデルの基本コンセプト，理論，仮説，それらの関係.
	Emergence 創発	Agentの適応的な特性や行動により生じる（マクロな）結果.
	Adaptation 適応	Agent自身や環境変化に反応して意思決定し行動を変更するルール.
	Objectives 目標・対象	Agentが追求する性向の定義，目標の測定.
	Learning 学習	Agentが経験を通じて特性を変化させる方法.
	Prediction 予測	Agent自身や環境の将来の状態に対する考え方.
	Sensing 感知	意思決定において参照する内部および外部の状態変数.
	Interaction 相互作用	Agent間の直接・間接の相互作用.
	Stochasticity 偶然性	モデル上のプロセスにおいてランダムの前提が置かれている箇所.
	Collectives 集団	Agentが形成している，所属している集団.
	Observation 観察	検証や分析のためにモデル（シミュレーション）から得られるデータ.
Detail	Initialization 初期化	モデルの初期状態.
	Input data インプットデータ	モデルに対する外的なデータソース等からの入力.
	Submodels サブモデル	プロセス、スケジューリングにおけるサブモデル.

ODD プロトコルの項目とログの対応づけ

“モデルが扱っている現象を示すプロセス”は ODD プロトコルの「Entities, state variables, and scales」,「Emergence」,「Observation」から参照する．“エージェントの行動”は ODD プロトコルの「Entities, state variables, and scales」,「Process overview and scheduling」,「Observation」から参照する（表 3-2).

ここで，扱う 4 つの項目の ODD プロトコルでの説明は表 3-1 から次のようになっている．

「Entities, state variables, and scales」：モデルに含まれるエンティティ（エージェントや環境)

とそれらを特徴付ける変数や特性.「Emergence」：エージェントの適応的な特性や行動により生じるマクロな結果,「Process overview and scheduling」：シミュレーションのプロセス.（エージェントがどういった順番に何を行うか)「Observation」：検証や分析のためにシミュレーションの結果から得られるデータ.

これらの項目を用いて 2 つのログを同定する．扱う現象についての発生プロセスについてはどのようなエージェントにより「Entities, state variables, and scales」，どのような現象が創発するか「Emergence」，さらにそれはどの要素で観測可能か「Observation」というように同定する．エージェントの行動はどのようなエージェントが「Entities, state variables, and scales」，どのように行動するか「Process overview and scheduling」，それはどの要素で観測可能か「Observation」というように同定する.

表 3-2 各ログと ODD プロトコルの対応

	階層的クラスタリング	決定木分析
1. Purpose		
2. Entities, state variables, and scales	✔	✔
3. Process overview and scheduling		✔
Design concepts		
4. Design concepts		
• Basic principles		
• Emergence	✔	
• Adaptation		
• Objectives		
• Learning		
• Prediction		
• Sensing		
• Interaction		
• Stochasticity		
• Collectives		
• Observation	✔	✔
Details		
5. Initialization		
6. Input data		
7. Submodels		

3.3.2 ログから規則性を抽出するアルゴリズム

本節ではもちいたアルゴリズムについての説明を行う，扱う現象についての発生プロセスに対しては，時系列データのみの場合（本章内 デモンストレーション，第 4 章 適用例 1）は距離関数として，動的時間伸縮法を用いた．動的時間伸縮法は単純なユークリッド距離とは異なり，“時系列の形”を比較可能である．時系列データでは表現できない場合（第 4 章 適用例 2）は，ログを時間軸にそって分析者が決めるルールベースで文字列に変換し，レーベンシュタイン距離を用いた．レーベンシュタイン距離は 1 文字の挿入・削除・置換によって，一方の文字列をもう一方の文字列に変形するのに必要な手順の最小回数で定義され，発生したことの順序を分類できる．エージェントごとのステップごとの行動をリスト構造として，C4.5 の説明変数として用いる．

また，手順全体の疑似コードで図 3-3 に示す．この手順の python による実行を付録に記載する．ただし，モデルによって異なるモデル・ログの箇所は記載しない．

```
propposed_method(){

//モデル・ログ
 param:対象とするパラメータセット
 N:試行回数
 log1:モデルで扱う現象の発生プロセス（ODDプロトコルにより同定）
 log2:エージェントの行動（ODDプロトコルにより同定）

 log1,log2 <- simulation_model(param, N)

//処理
 //クラスタリング
 labels:ログごとのクラスター番号
 labels <- hierarical_clustering（log1）

 //決定木分析
  tree:決定木
  tree <- decision_tree（labels，log2）

//アウトプット
 return labels, tree

}
```

図 3-3 提案手法の手順（疑似コード）

利用したアルゴリズム

本節では，本研究で利用したアルゴリズムについて説明する．まず，ログに含まれるふるまいを発見するアルゴリズムとして階層的クラスタリングを利用している．クラスター間の距離を計算する方法として ward 法を利用している. また, クラスター数の決定方法として, Calinski-Harabasz(CH)基準を利用している．さらに，類型ごとのエージェントの行動についての特徴を抽出するために，決定木分析を行っている．アルゴリズムとして，最も典型的な決定木手法の 1 つである C4.5 を利用している．

階層的クラスタリング

階層的クラスタリング最も似ている組み合わせから順番にまとまり（クラスター）にしていく方法である(Everitt et al. 2011)．階層的クラスタリング(Hierarchical clustering)は，クラスターの階層を構築しようとするクラスタリングの方法である．階層的クラスタリングのアプローチは，一般的に凝集型と分岐型の 2 つのタイプに分類される(Murtagh and Contreras 2012)．凝集型（agglomerative）はボトムアップなアプローチであり，各データは単独のクラスターで開始し, 階層の上に移動するにつれてデータをマージしていく. 分枝型（divisive)はトップダウンなアプローチであり，すべてのデータが 1 つのクラスターで開始され，分割が階層を下っていくにつれて再帰的に分割が実行される．

本研究では前者の凝集型を採用している．この手法は，1 個の対象だけを含む N 個のクラスターがある初期状態から，クラスター間の距離（非類似度）関数に基づき，最も距離の近い二つのクラスターを逐次的に併合する．そして，この併合を，全ての対象が一つのクラスターに併合されるまで繰り返すことで階層構造を獲得する．この階層構造はデンドログラムによって表示する．クラスター間の距離関数としては最短距離法，最長距離法，群平均法，ward 法などいくつか提案されているが本研究では ward 法を用いてる．ward 法のクラスター間の距離関数の定義は式(3.4)で示している．

最短距離法（nearest neighbor method）または 単連結法（single linkage method）

$$\mathrm{D}(C_1, C_2) = \min_{x_1 \in C_1, x_2 \in C_2} \mathrm{D}(x_1, x_2) \qquad (3.1)$$

最長距離法（furthest neighbor method）または 完全連結法（complete linkage method）

$$\mathrm{D}(C_1, C_2) = \max_{x_1 \in C_1, x_2 \in C_2} \mathrm{D}(x_1, x_2) \qquad (3.2)$$

群平均法（group average method）

$$\mathrm{D}(C_1, C_2) = \frac{1}{n_1 n_2} \sum_{x_1 \in C_1} \sum_{x_2 \in C_2} \mathrm{D}(x_1, x_2) \qquad (3.3)$$

ward 法（Ward's method）

$$\mathrm{D}(C_1, C_2) = \mathrm{E}(C_1 \cup C_2) - \mathrm{E}(C_1) - \mathrm{E}(C_2) \qquad (3.4)$$

$$\text{ただし，} \quad \mathrm{E}(C_i) = \sum_{x \in C_i} \left(d(x, c_i)\right)^2$$

$$\text{ここで} c_i \text{は} C_i \text{の重心であり，} \quad c_i = \sum_{x \in C_i} x / |C_i|$$

Calinski-Harabasz(CH)基準

クラスタリングにおいて，クラスターの数をどう決定するかという問題は，ヒューリスティックに依存し，最適なクラスター数の基準としていくつかの方法がある．本研究では，Calinski-Harabasz（CH)基準を採用している(Caliñski and Harabasz 1974)．CH 基準では正方形のクラスター内距離の合計，つまり，クラスター内のデータの凝集性を用いてる．さらに，Calinski-Harabasz（CH)基準は，クラスター内の凝集性に加えて，クラスター間の離散性を考慮する．したがって，本研究では Calinski-Harabasz（CH)基準を採用している．

クラスター数の決定は PseudoF が最大となるように決定される．PseudoF は，クラスター間分散（全データセットの重心からのすべてのクラスター重心の分散）とクラスター内の分散（各クラスタ内の分散の平均）の割合で定義され（3．5）式によって計算される．

$$\mathrm{PseudoF} = \frac{(T - W_k)/(k-1)}{W_k/(n-k)} \qquad (3.5)$$

T: 全サンプルの距離二乗和

W_i: クラスター内距離二乗和

k: クラスター数

n: 全サンプルの数

距離関数

本研究では階層的クラスタリングを適用するにあたって，ログ間の距離関数を定義することを前提としている．階層的クラスタリングにおいては，データ間の距離がクラスタリングの結果を規定する．典型的なクラスタリングの場合，1 次元の配列間のユークリッド距離を用いる．これは本研究においては，1 試行のログを 1 次元の配列で表現することに相当する．

一方，モデルで扱う現象の発生プロセスは時系列データであるため，必ずしも 1 次元の配列として表現することが適切でない場合が多い．そのため，本研究ではログ間の距離の計算方法として，動的時間伸縮法と，レーベンシュタイン距離を用意し，モデルで扱う現象の発生プロセスの形式に適した距離を利用している．

動的時間伸縮法

時系列解析において，動的時間伸縮法（DTW：Dynamic time-wrapping）は，2 つの時系列間の類似度を計算するアルゴリズムの 1 つである(Jeong Y, Jeong M, and Omitaomu 2011), (Berndt and Clifford 1994)(Senin 2008)．DTW では 2 つの時系列の観測点の最も距離が最短の対応付けをとることで形が比較可能である(図 3-4)．

また，DTW の特徴として，時系列データの長さが異なる可能性のある場合も比較可能である．例えば，歩行の動きの類似性を検出しようとした場合，一方の人が他方よりも速く歩いていたとしても，あるいは観測中に加速や減速があったとしても，DTW で検出することができる．それゆえ，DTW は，動画，音声データなどに用いられることが多い(Jeong Y, Jeong M, and Omitaomu 2011)．

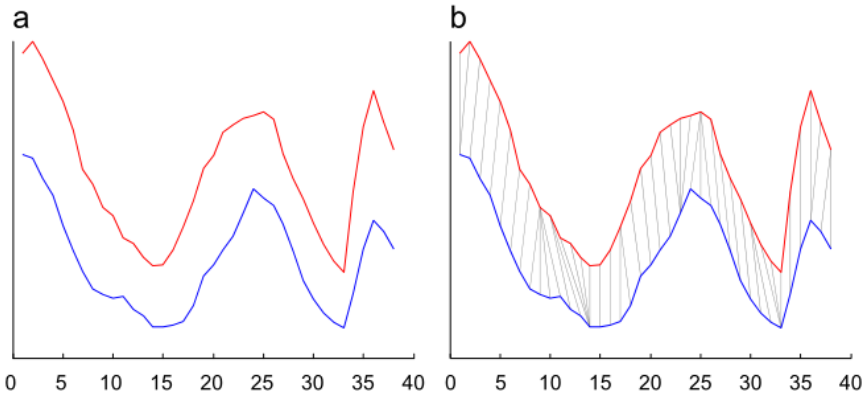


図 3-4 DTW では 2 つの時系列の観測点の最も距離が最短の対応付けをとることで形が比較が可能(Jeong Y, Jeong M, and Omitaomu 2011))

疑似コードは以下である．

```
DTWDistance(s: array [1..n], t: array [1..m]) {
  DTW := array [0..n, 0..m]

  for i := 1 to n
    DTW[i, 0] := infinity
  for i := 1 to m
    DTW[0, i] := infinity
  DTW[0, 0] := 0

  for i := 1 to n
    for j := 1 to m
      cost := d(s[i], t[j])
      DTW[i, j] := cost + minimum(DTW[i-1, j  ],
                                  DTW[i  , j-1],
                                  DTW[i-1, j-1])

  return DTW[n, m]
}
```

レーベンシュタイン距離

レーベンシュタイン距離は，二つの文字列がどの程度異なっているかを示す距離の一種である．レーベンシュタイン距離は，1 文字の挿入・削除・置換によって，一方の文字列をもう一方の文字列に変形するのに必要な手順の最小回数として定義される(Gusfield 1997)(Sebastiani F 2002)．詳細には以下の式によって定義される．

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) \\ min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1 \end{cases} \end{cases} \quad (3.6)$$

動的計画法により解かれるのが一般的であり以下はその疑似コードである．

```
LevenshteinDistance(const char *s, int len_s, const char *t, int len_t)
{
  int cost;

  if (len_s == 0) return len_t;
  if (len_t == 0) return len_s;

  if (s[len_s-1] == t[len_t-1])
     cost = 0;
  else
     cost = 1;

  return minimum(LevenshteinDistance(s, len_s - 1, t, len_t   ) + 1,
                 LevenshteinDistance(s, len_s   , t, len_t - 1) + 1,
                 LevenshteinDistance(s, len_s - 1, t, len_t - 1) + cost);
}
```

C4.5 アルゴリズム

C4.5 は，決定木を生成するために使用されるアルゴリズムである(Quinlan 1993,1996)．決定木は予測モデルであり，ある事項に対する観察結果から，その事項の目標値に関する結論を導く．C4.5 によって生成された決定木は分類問題に使用することができる．

決定木では分岐の基準をどうするかが最大の重点となるが, C4.5 ではエントロピー（情報量）を用いて木を分岐させる条件を決定する．あるデータ集合の状態 X に関する「あいまいさ」は以下の式で定義されるエントロピー Info(X)で測ることができる.ある値で木を分岐させたとき, 分岐前の状態をX_0，分岐後の状態をX_1とすると，Gain（相互情報量）は(3.8)式で計算され，最もGainが大きくなる分割が選択さる．これを再帰的に繰り返すことで，決定木を得る．

$$\mathbf{Info(X)} = -\sum_{j=1}^{k}\{p(j|t)log_k p(j|t)\} \quad (3.7)$$

$$\mathbf{Gain} = \mathbf{Info}(X_0) - \mathbf{Info}(X_1) \quad (3.8)$$

3.4 分居モデルへの適用による手法のデモンストレーション

本節では本研究でエージェントシミュレーションのログから抽出している規則性である PSP-AA が具体的にどのようなものでどのような手順で抽出されているのかを具体例をもって示す.

分居モデルは Shelling(1971)によって提案された，エージェントシミュレーションのモデルであり当該のモデルを対象として数多くの研究が行われている．入力パラメータを変化させどのように分居するかについてを扱った研究(Singh, Vainchtein and Weiss 2009) (Domic, Goles and Rica 2011)．分居モデルを物理モデルなどで近似的にモデル化し，同様の挙動を示すモデルを構築する研究(Vinković and Kirman 2006)(Gauvin, Vannimenus and Nadal 2009)．分居モデルを拡張し，応用的な目的に適用しようとする研究(Crooks 2010)などがあげられる．しかし，分居モデルのような古典的なモデルであってもプロセスについての規則性は十分に研究されていない.

3.4.1 分居モデルとその挙動

分居モデルは，セル上の空間に配置された 2 種類の複数のエージェントが自分と異なるエージェントから離れるというルールだけで同種のエージェントに分かれる現象をモデル化したものであり，人種の分居などの説明に用いられる.

しかし，多くのエージェントシミュレーションと同様に，同じ初期配置でも乱数の初期値によってエージェントの行動が異なり，それにより収束パターン（評価値の時間変化や収束時間=評価値の時系列）が異なる．本研究ではエージェントの行動と収束パターンの関連を詳しく調べたいが (ある収束パターンはどのようなエージェントの行動が原因かなど)，個別試行のログ分析を行ってもログが細かすぎるため，規則性を探すのが困難である.

各エージェントの満足度は，周囲の同色のエージェントの割合により(3.9)式で決定され，一定割合（今回は閾値 50%）より小さい場合不満足となる．不満足なエージェントの割合(3.10)で計算され, 不満足なエージェントの割合がゼロになったとき, シミュレーションが終了する.

満足度 = 周囲(8方向1マスの範囲)に存在する同色エージェントの数 ÷
周囲に存在するエージェントの数　　(3.9)

不満足なエージェントの割合 = 不満足なエージェント数 ÷ 全エージェントの数　　(3.10)

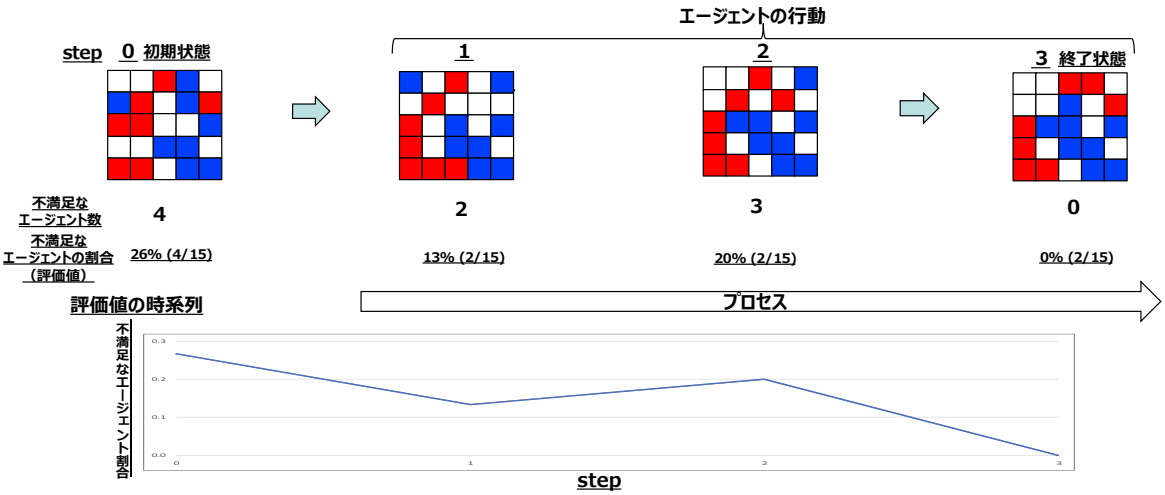


図 3-5 分居モデルの挙動の例

挙動の一例を，図 3-5 に示す．初期状態では，不満足なエージェントが 4 体存在するため，不満足なエージェントの割合は 4/15(≒26.7%)である．エージェントが移動するに従い，不満足なエージェントの数が変化していき，評価値である不満足なエージェントの割合も変化していく．Step4 で不満足なエージェントがゼロになり，不満足なエージェントもゼロになっていることが確認でき，この段階でシミュレーションが終了となる．

分居モデルの挙動は試行のたびに異なる収束パターンを示す（図 3-6）．エージェントの移動先はランダムに決定されるため，試行のたびにエージェントの動きが異なる．エージェントの動きが異なると，他のエージェントからは周囲のエージェントが変わるということなので，不満足なエージェントも試行によって変化する．このことにより，不満足なエージェントの割合も変化する．そのため，エージェントがランダムに動くことが原因で，収束 Step や評価値の時系列の形が変わってくる．

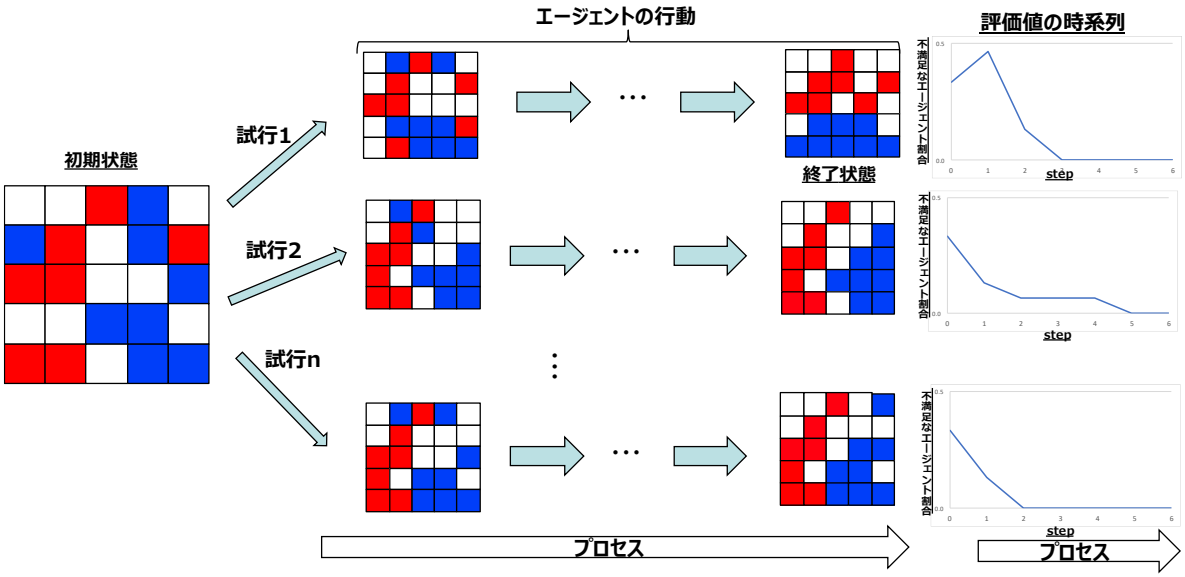


図 3-6 試行のたびに収束パターンは異なる.

3.4.2 提案手法の適用

提案手法で PSP-AA を得ることができるかを確認するために，分居モデルへ提案手法の適用を行った．分居モデルとは異なる人種間が分居する現象を説明する古典的なモデルである．エージェントは周囲のセルに存在する同色の割合で満足度が決まり，満足度が閾値以下であれば他のセルに移動する．今回は，理解のしやすさため 5×5 のセルとし，初期状態を固定した（図 3-5 初期状態）．全セルに占めるエージェントの数は 15 体，移動を行う不満足の閾値は 50%とした．

分居モデルは，異なる初期パラメータで結果がどう変化するかの分析が広く行われている(Singh and Vainchtein & Weiss 2009)．本研究では，固定された初期パラメータにおいて，異なる結果を得られるか確認するため，クラスター数は 2 とした．ODD プロトコルによる記述は Railsback and Grimm(2011)を利用した．ログ数は 4000 件である．ログ間の距離は不満足なエージェントの割合の時系列を動的時間伸縮法(Dynamic Time Warping)での距離とした．動的時間伸縮法の window size の設定は行っていない．クラスタリングは階層的クラスタリングの ward 法を利用しており，決定木は C4.5 を利用している．

“モデルが扱っている現象の発生プロセス”と“エージェントの行動”は(Railsback and Grimm 2011)での ODD プロトコルによる記述を利用している（図 3-7）．具体的には前節の挙動説明で利用していた不満足なエージェントの時系列と，エージェントがどのセルに存在するかについてのログを対象としている．(Railsback and Grimm 2011)での ODD プロトコルによる記述では，Entities, state variables, and scales の項目は，エージェントは家庭であり，セル上の位置と色で特徴づけられるとある．Emergence の項目は分居のパターンが創発すると Process

overview and scheduling の項目は不満足なエージェントは移動し，満足度を更新し，全員が満足となったら終了とある．これらの観測項目を指定する Observation の項目には，“それぞれのセルにどのエージェントが存在するか”，“不満足なエージェントの割合”，“全エージェントの満足度の平均”の 3 つの観測するデータが記載されている．このうち，“不満足なエージェントの割合”，“全エージェントの満足度の平均”はそれぞれの定義上，強い逆相関関係にあるため，どちらかを用いる場合と，両方を用いる場合の情報量に大きな差はないと考え“不満足なエージェントの割合”のみを扱っている．

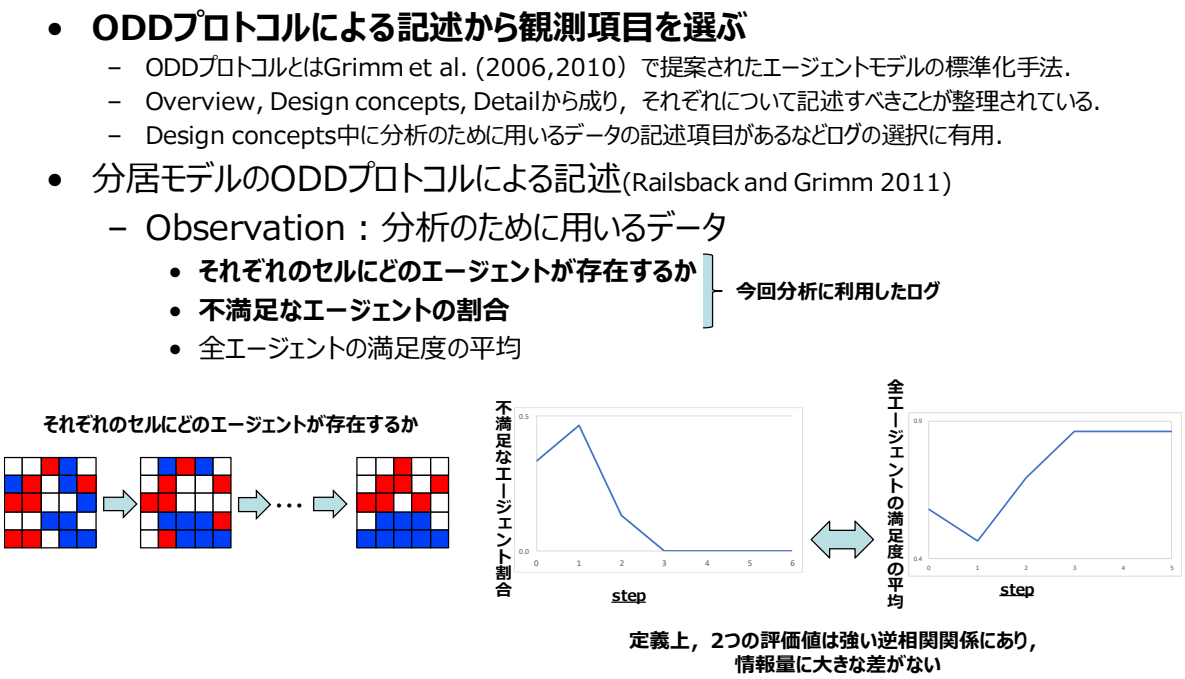


図 3-7 ODD プロトコルから観測ログを同定

用いたログのデータ構造を説明する．不満足なエージェントの割合の時系列については，（図 3-5）（図 3-6）からも明らかにわかるように，時系列データであるが，エージェントがどのセルに存在するかについては表 3-3 のようなデータを決定木の説明変数として用いている．具体的には Step ごとにセル数分の項目を用意する．セル数×ステップ数の項目数となる．今回は Step3 までのデータを用いており，75 項目(=3×25)となる．

表 3-3 決定木に入力したエージェントの行動.

	説明変数										被説明変数
	step1						…	step3		…	分類結果
	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(2,1)	…	(1,1)	(1,2)	…	
ログ1	青色	青色	赤色	青色	空白	空白		空白	空白		類型2
ログ2	空白	青色	赤色	空白	青色	空白		空白	青色		類型2
ログ3	青色	青色	赤色	空白	空白	空白		空白	空白		類型2
ログ4	空白	青色	空白	空白	空白	空白	…	青色	空白	…	類型2
・ ・ ・											・ ・ ・
ログ4000	空白	青色	空白	赤色	青色	空白		青色	赤色		類型1

適用結果

クラスタリングの結果を図 3-8, 3-9 に示す. 得られた 2 つのクラスターは評価値が一旦上昇する類型と, 徐々に収束していく類型と解釈できることが図 3-8 からわかる. 図 3-8 左では全ログから 30 ログをランダムで抽出し示している. この状態では, 評価値が徐々に収束していっている様子しか観測するのは難しい. クラスタリングを行うことで, ログが特徴を得られていることが分かる.

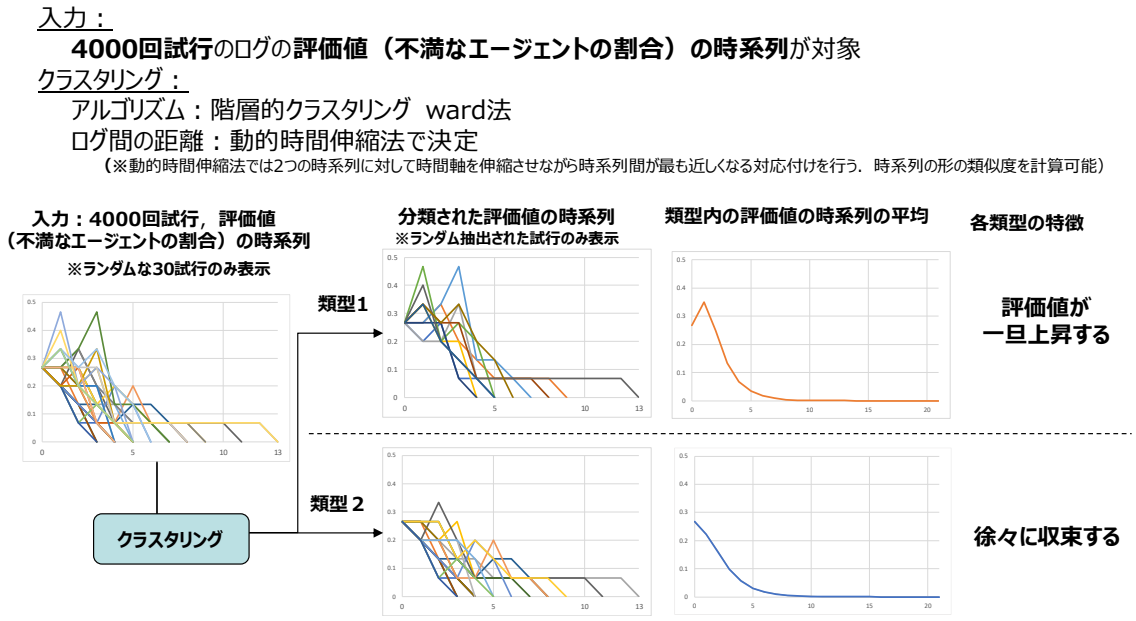


図 3-8 2 つに類型化された実際のログ.

図 3-9 では特定の指標で，クラスタリングされた結果をマッピングしている．図にある cluster_a が類型 1，cluster_b が類型 2 に相当しているが，マッピングしている指標だと 2 つの Cluster の特徴をうまく分離できていないことが確認できる．つまりログの分布では特徴を示すことが難しいことが分かる．

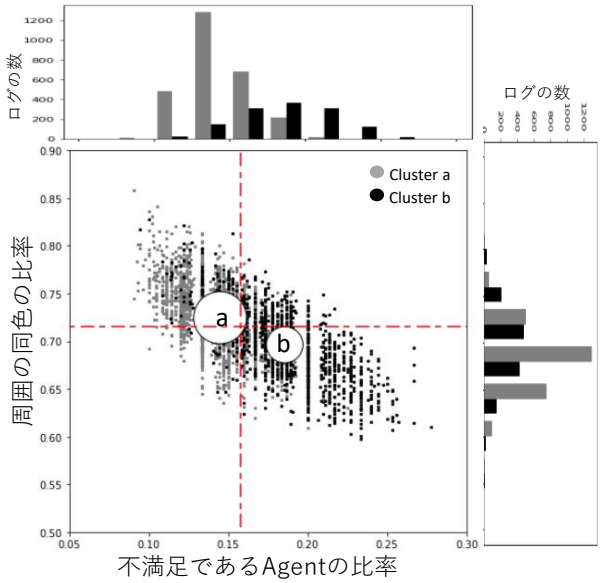
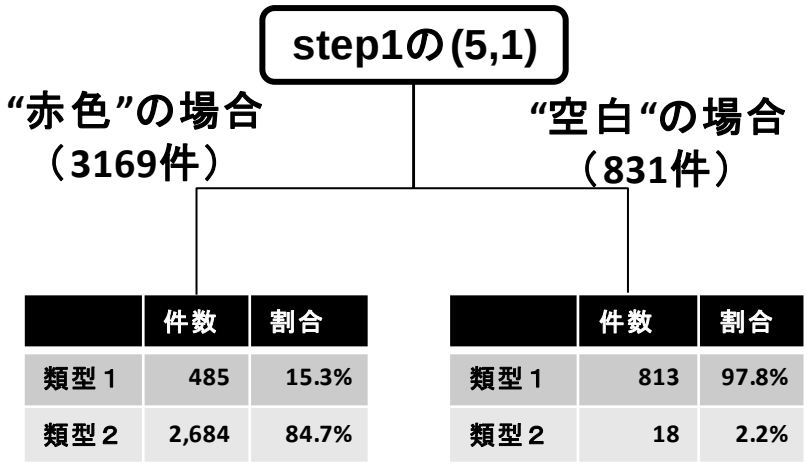


図 3-9 分居モデルのログから得られた 2 つのクラスター． 2 つのクラスターの分布には重複がある．

決定木分析の適用結果を図 3-10 に示す．Step1 のセル(5,1)に存在するエージェントによって類型 1 と類型 2 は説明できることが分かる．図 3-10 より，Step1 の(5,1)の赤色エージェントが動いた場合， 97.8%（813/831）で類型 1 に，動かない場合 84.7%（2684/3169）で類型 2 になることが読みとれる．



	件数	割合
類型１	485	15.3%
類型２	2,684	84.7%

	件数	割合
類型１	813	97.8%
類型２	18	2.2%

図 3-10 決定木分析によって得られた結果.

3.4.3 得られた規則性について

分居モデルについて，特定の初期状態からのログについて，特定のエージェントの行動によって，特徴の異なる 2 つの類型に分けることができる例を示した.

この事例に示すような「シミュレーションのプロセスについての類型と，その類型と関連があるエージェントの行動(Patterns of Simulation Processes and Actions of Agents.)」を本研究では“PSP-AA”と呼び，ABS のログから”PSP-AA”を抽出する手法の提案を行い，既存モデルに適用し得られる規則性の妥当性と新規性を示す.

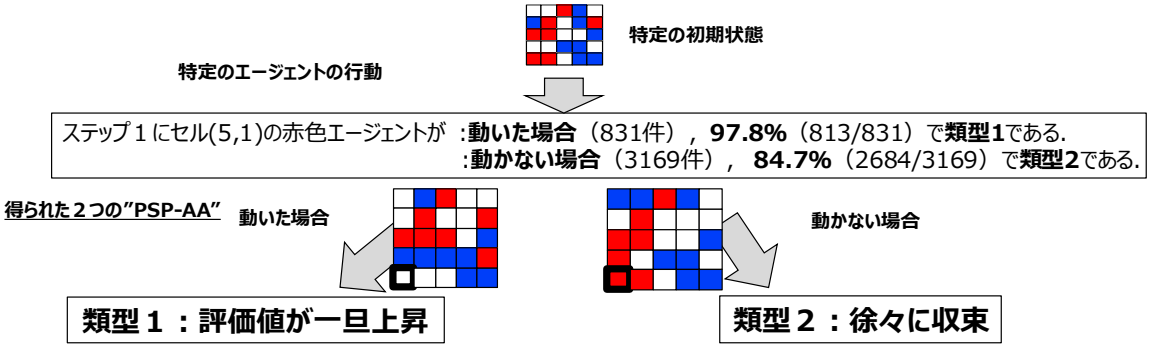


図 3-11 分居モデルから得られた PSP-AA

第 4 章　提案手法の適用

4.1 はじめに

本章では，提案手法を既存のモデルに適用する．具体的に，組織シミュレーション(Kobayashi et al. 2013)と，金融シミュレーション(菊地ら 2016)への適用を行う．本研究で提案する抽出手法を 2 つの既存のエージェントシミュレーションに適用することで手法によって得られる規則性の妥当性・新規性を確認している．得られた規則性について，既存のエージェントシミュレーションを用いた既存研究での結果と比較し，整合的であることをもって妥当であることを確認する．また，既存研究では得られていない規則性が得られたことをもって，新規性があること確認する． 手法を適用した結果，1 つ目の適用例である改善・逸脱モデルでは既存研究の主要な結論が，本研究で得られた規則性からも示せた．また，既存研究では論じられていない改善・逸脱のプロセスについての新たな類型を得た．2 つ目の適用例である連鎖破綻モデルについて，既存研究ではモデルにおいて連鎖破綻が発生することを示すにとどまっていたのに対して，本研究では連鎖破綻のプロセスについての 3 つの類型を示している． 対象の 2 つのモデルを選んだ理由として，これらの研究では，特定試行のログ分析による，ある程度詳細な分析が行われており，分析結果の比較が可能であるためである．

4.2 適用 1：組織シミュレーションへの適用

本節においては，既存の組織シミュレーションモデル(Kobayashi et al. 2013)である改善・逸脱モデルを例題とし，提案手法を適用する．まず，改善・逸脱モデルについての概要と，先行研究における分析内容，今回のモデルの設定を説明し，その後提案手法を適用する．

4.2.1 先行研究：改善・逸脱モデル

改善・逸脱モデルは組織における改善と逸脱を同じ原理で説明できるシミュレーションモデルである．(Kobayashi et al. 2013)では社会規範に違反した組織的な行為である組織体逸脱（宝月 2004）と，企業組織において取り組まれる生産性向上活動である改善（Imai 1986）とを，既定の業務方法を改変する意味において，規範からの逸脱の一種と捉えられると考え，両者を組織や社会にどのような効用を与えるかによって定義した(図 4-1)．つまり，社会効用が上がり，組織効用が上がる場合は改善であり，社会効用が下がり，組織効用が上がる場合は逸脱である．改善・逸脱モデルの目的は，改善逸脱に至る過程・要因を分析することである．エージェントは，組織の構成員である．エージェントは自身の満足度を向上させるため，自身で行動を探索（探索学習），または他者の行動を模倣（模倣学習）する．エージェントの行動によって，社会効用・組織効用が決まる．さらに，行動が繰り返されることで，社会効用・組織効用が変化していき改善や逸脱に至る．モデルの ODD プロトコルによる記述を表 4-1 に示す．

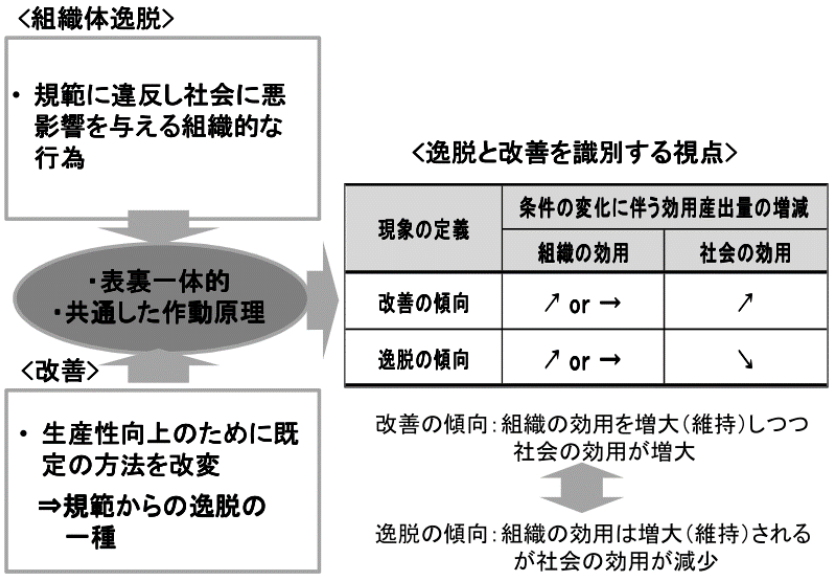


図 4-1 改善・逸脱の定義(Kobayashi et al. 2013)

このモデルを用いて，Kobayashi et al.(2013)は組織の多様度と，改善・逸脱との関連を分析している．手法は第二章で示したマクロレベルの分析と，ミクロレベルの分析を行っている．マクロレベルの分析においては多様度を変化させて，出力される社会・組織効用の分布の変化を，分布単位で観測している．その結果，多様度が大きくなるについて社会効用も大きくなるが，組織効用は変化しないという結果を得ている．それによって，多様度が高まるほど改善の傾向が高まることを主張している．

一方，ミクロな分析においてはログを追跡できるよう，モデルを小さくし，特定のログを抽出して，その挙動が現実的なケースとして記述できるかを確認している．エージェントを 3 体とし，分析者が分析しやすくしている．組織は一様な組織と多様な組織について行っている．この一様・多様は構成員の個人効用の関数によって定義される．その結果，一様な組織から逸脱と解釈できるログを抽出し，ビジネスケースとして記述している．同様に，多様な組織からは改善と解釈できるログを抽出し，ビジネスケースとして記述している．それによって，改善・逸脱モデルの妥当性を主張している．先行研究では，改善・逸脱モデルを用いて，多様度が高まるほど改善の傾向が高まることを示すとともに，一様な組織から逸脱のビジネスケースを，多様な組織からは改善のビジネスケースを得ている．

本研究では，(Kobayashi et al. 2013)が個別ログの分析で用いたモデルの設定を利用する．Agent は 3 体である．表 4-1 にある友人ネットワークは固定しており完全グラフである．(Kobayashi et al. 2013)では一様な組織・多様な組織という 2 つの初期パラメータに対する分析を行っている．社会効用と組織効用には相反を持たせており，2 つの組織ではエージェントが持つ個人効用関数のみが異なる．一様な組織においては，3 体のエージェントとも，組織効用

を優先する効用関数を持ったエージェントである．一方，多様な組織においては，エージェント 1 は社会効用を優先する個人効用関数を持ち，エージェント 2 は組織効用を優先する個人効用関数を持ち，エージェント 3 は両方を同等に優先する個人効用関数を持つ．

表 4-1 改善・逸脱モデルの ODD プロトコルによる記述．(Kobayashi et al. 2013)より作成.

	要素	説明
Overview	Purpose 目的	不完全情報のもとでのAgentの行動によって改善と逸脱が創発する過程の分析． パラメーターの変化に伴う改善と逸脱が生ずる過程とその要因を考察．
	Entities, state variables, and scales エンティティ， 状態変数，スケール	Agentである組織の構成員にとっての不完全情報環境として，社会効用関数，組織効用関数，さらにAgentごとの変数である個人効用関数の3階層のランドスケープを設定する．社会効用関数は，例えば消費者のニーズであり，Agentの行動で社会にもたらす効用が決定．組織効用関数は，企業のビジネスモデルなどであり，Agentの行動で組織にもたらす効用が決定．個人効用関数は，現在の行動がどの程度自身にとって快適な状態にあるのかを決定する．
	Process overview and scheduling プロセス， スケジューリング	Agentは自身の満足度を向上させるため，自身で行動を探索（探索学習），または他者の行動を模倣（模倣学習）する．しかし満足度が向上しない場合，現状維持する．探索か模倣かは確率的に決定される． 次の1～3を1Stepして繰り返す．1.Agentはそれぞれが自身の行動に関する意思決定を行う．2.すべてのAgentの行動により，社会効用，組織効用が決定される．3.組織効用により報酬が決まり各Agentにフィードバックされる．
Design concepts	Basic principles 基本原則	階層的なランドスケープ：　社会，組織，個人の3階層のランドスケープ 社会・組織効用がともに上昇する現象を改善と見なし，社会効用は上昇するが組織効用が下降する現象を逸脱と見なす．（図５）
	Emergence 創発	階層間の算出効用量の関係の変化
	Adaptation 適応	行動状態の変化：Agentは自身の満足の向上を目指して行動を変化
	Objectives 目標・対象	満足度の構成：社会効用，報酬，個人効用の総和
	Learning 学習	①個人の探索による学習 ②友人の模倣による学習
	Prediction 予測	－
	Sensing 感知	不完全情報による選択 ①自身の行動状態の周辺効用の増減 ②友人の行動状態 ③友人が得た報酬
	Interaction 相互作用	ネットワークと通じたAgent間の行動状態と報酬情報の交換
	Stoohasticity 偶然性	友人選択：インフォーマルネットワークは，ランダムに選択されたAgent間に形成
	Collectives 集団	効用配分：組織効用の産出量に応じた効用配分 集団を構成するAgentの個人効用関数の多様性
	Observation 観察	①社会効用の産出量の変化の過程 ②組織効用の産出量の変化の過程 ③個人効用の産出量の変化の過程

4.2.2 実験の設定

本節では，改善・逸脱モデルに提案手法を適用する．具体的に，“モデルが扱っている現象の発生プロセス”のログとして社会効用・組織効用の時系列をクラスタリングすることで，ログの類型化を行う．（手順 1：クラスタリングの適用）．また，類型の”エージェントの行動”を決定木分析により抽出する（手順 2：決定木による要因分析の適用）．

手順 1：クラスタリングの適用の設定

手順 1:クラスタリングを適用した際の，データとアルゴリズムの概要を表 4-2 に示す．（データ 1)ログの生成条件は改善・逸脱それぞれの初期パラメータでの実行から得られた 1000 試行ずつのログを利用している．本研究で得らえた結果と先行研究での結果を比較可能にするために，用いているパラメータは(Kobayashi et al. 2013)と同一である．（データ 2)ODD プロトコルで記述されたモデルから同定された“モデルが扱っている現象の発生プロセス”のログとして，社会効用の産出量の変化の過程，組織効用の産出量の変化を扱っている．(アルゴリズム) クラスタリングのアルゴリズムは階層クラスタリングの ward 法を利用し，ログ間の距離は動的時間伸縮法で計算する．今回は時系列が 2 つであるため，2 つの和を距離としている．クラスター数を 3 としているが，これは Kobayashi et al. (2013)では改善のログ，逸脱のログと異なる 2 つのログを得ていることから，新たな性質の類型を得るために必要な，最小限のクラスター数である 3 とした．

表 4-2 手順 1：クラスタリングを適用した際の，データとアルゴリズムの概要

データ 1 ：ログの生成条件	改善・逸脱それぞれの初期パラメータでの実行から得られた 1000 試行ずつのログ．Kobayashi et al. (2013)で用いられているパラメータと同一
データ 2 ：“モデルが扱っている現象の発生プロセス”のログ	社会効用の産出量の変化の過程，組織効用の産出量の変化(ODD プロトコルによる記述より同定)
アルゴリズム	階層的クラスタリング ward 法 ログ間の距離：動的時間伸縮法 クラスター数：3

Kobayashi et al. (2013)の研究では，多様な組織と一様な組織について分析しており，本研究でも同様に行う．多様な組織であるか，一様な組織であるかは，シミュレーションの初期パラメータで定義されており，多様な組織のパラメータ，一様な組織のパラメータについてログの出力を行う．それぞれのパラメータで 1000 回試行を行い，これらのログを多様な組織，一様な

組織それぞれのログ全体とする．

また，Kobayashi et al. (2013)の ODD プロトコルによるモデル記述から扱うログを決定している．“モデルが扱っている現象の発生プロセス”のログは，社会効用の産出量の変化の過程，組織効用の産出量の変化の過程である．Kobayashi et al. (2013)において，観測の項目は社会効用の産出量の変化の過程，組織効用の産出量の変化の過程，個人効用の産出量の変化の過程と記述されている（表 4-1）．改善・逸脱モデルにおいて，組織が改善の結果になるか，逸脱の結果になるかが焦点であるため，組織効用と社会効用の変化で改善・逸脱は定義されるため今回はその範囲を扱っている．

図 4-3 は扱っているログ(社会効用と組織効用の時系列)の例である．図 4-3 は組織効用と，社会効用の推移を示しており，横軸がシミュレーションの Step 数，縦軸がそれぞれの効用の大きさを示している．シミュレーションでは Agent である組織の構成員の Step ごとの意思決定によって図 4-3 のように，組織効用と社会効用が変化していく．

クラスタリング手法は，階層的クラスタリング手法の ward 法を利用しており，ログ間の距離は時系列データであるため，時系列データの距離を求める手法である動的時間伸縮法を利用している．動的時間伸縮法の window size の設定は行っていない．

手順２：決定木分析の適用の設定

手順２：決定木分析の適用した際の，データとアルゴリズムの概要を，表 4-3 に示す．

（データ 1）ログの生成条件は手順１と同様であるが，先行研究で得られていなかった類型である類型 2 の発生要因を分析するために，類型 1，2 のみに含まれるログのみを対象とする．（データ 2）“エージェントの行動”のログは，3 体の Agent それぞれが Step ごとの意思決定である探索，模倣，現状維持の内どの意思決定を行ったかのログを利用している．（データ 3）それぞれのログがどの類型に属すかは手順１で得られたログごとの属す類型である類型１または類型 2 である．（アルゴリズム）決定木による要因分析のアルゴリズムは，C4.5 を利用している．

表 4-3 利用したデータとアルゴリズム（改善・逸脱モデルへの適用の手順 2）

データ１：ログの生成条件	手順１と同様であるが，手順１で得られた類型１，２のみを対象としている．計 914 件
データ２：“エージェントの行動”のログ	Agent・Step ごとの行動である探索・模倣・現状維持についてのログ
データ３：それぞれのログがどの類型に属すか	手順１で得られたログごとのラベル
アルゴリズム	C4.5

表 4-4 決定木分析で用いたエージェントの行動ログの例.

Agent	各 Step の意思決定			
	Step2	Step3	Step4	Step5
Agent1	探索	探索	模倣	維持
Agent2	探索	探索	模倣	維持
Agent3	模倣	模倣	模倣	維持

ログの生成条件はクラスタリングで用いたものと同様であるが，対象は類型 1,2 を分ける要因を明らかにするのが目的であり，類型 3 を除外した．また，ログごとの属する類型は，クラスタリングで得られたそのログが，どの類型に属すかのラベルであり類型 1 または類型 2 である．計 914 件のログを交差検定(Kohavi, R. 1995)のため 680 件と 234 件に分け，それぞれ学習データ，評価データとした．

エージェントの行動は，3 体の各 Agent の Step ごとの意思決定である．意思決定は Agent ごとに Step2 以降，Step ごとに探索，模倣，現状維持のいずれかの意思決定を行う（表 4 - 4 ）．すべての Agent の意思決定により，社会効用，組織効用が決定され，それが報酬となって各 Agent にフィードバックされる．表 4- 4 は要因の候補となった意思決定のログの例である．どの探索を行うか，模倣を行うかは確率的に決定され，満足度が上昇しなかった場合，現状維持として行動を変化させない．

アルゴリズムは決定木分析の典型的な手法である C4.5 を利用した．データはそのログがどの類型に属すかが目的変数で, Agent・Step ごとにどの意思決定をとったかが説明変数である．このデータに C4.5 を適用する．

4.2.3 実験の結果

本節では，“モデルが扱っている現象の発生プロセス”のログとして社会効用・組織効用の時系列をクラスタリングすることで，ログの類型化を行う．（手順 1：クラスタリングの適用）．また，類型の”エージェントの行動”を決定木分析により抽出する（手順 2：決定木による要因分析の適用）．その結果，「組織効用を重視するエージェントの探索学習の回数が 5 回以上で，組織効用が一度上昇してから下がるという類型になる」という PSP-AA を抽出されたことを示す．

手順 1：クラスタリングの適用結果

図 4-2 は，多様な組織，一様な組織それぞれで手順 1 により得られたクラスターをプロットしたものである．図の横軸，縦軸である社会効用の変化，組織効用の変化について，モデルで発生する現象である改善と逸脱は，図 4-1 のように条件変化に伴う効用産出量の増減で定義される．定義に従い，条件変化を時間変化とし，社会効用の変化，組織効用の変化をそれぞれ社会効用, 組織効用の Step5 と Step2 時点の値の差分とした．つまり，社会効用の変化が正かつ，

組織効用の変化が正で改善の傾向，社会効用の変化が負かつ，組織効用の変化が正で逸脱の傾向となる．図 4 - 2 から読み取れるクラスターの性質は表 4 - 5 に記述する．

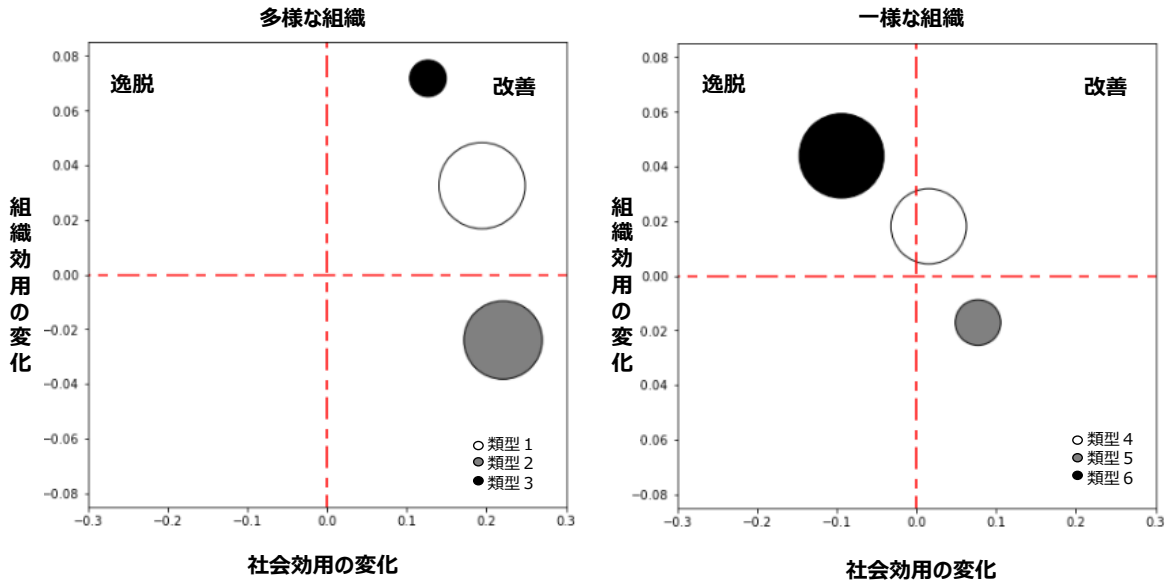


図 4-2 改善・逸脱モデルのログから手順 1 よって得られたログのクラスター： 先行研究(Kobayashi et al. 2013)で扱われている 2 つの異なる状況（一様な組織，多様な組織）からそれぞれ 3 つのクラスターを得た．円の面積はそのクラスターの頻度を示している．

表 4-5 図 4-2 から読み取れる類型ごとの性質．

パラメータ	試行回数	類型	各類型の性質			備考
			出現頻度	社会効用	組織効用	（先行研究との関係）
多様な組織	1000 回	類型 1	502 件	増加(+0.19)↗	増加(+0.04)↗	先行研究の結果に整合的．
		類型 2	412 件	増加(+0.22)↗	減少(-0.03)↘	提案手法により，初めて得られた．
		類型 3	86 件	増加(+0.12)↗	増加(+0.06)↗	先行研究の結果に整合的．
一様な組織	1000 回	類型 4	347 件	増加(+0.02)↗	増加(+0.02)↗	—
		類型 5	137 件	増加(+0.08)↗	減少(-0.02)↘	提案手法により，初めて得られた．
		類型 6	516 件	減少(-0.09)↘	増加(+0.04)↗	先行研究の結果に整合的．

表 4-5 は，図 4-2 にて示されている 2 組のクラスターの性質を，それぞれ表にまとめたものである．一様・多様な組織のログセットごとの 3 つクラスターごとに図 4-2 から読み取れる出現頻度，社会効用の変化，組織効用の変化を示している．この結果から，Kobayashi et al. (2013)に整合的な性質と，未知の性質を持つ類型を見出した．Kobayashi et al. (2013)における主要な結論の 1 つは"一様な組織では逸脱の傾向があり，多様な組織では改善の傾向がある"である．得られた類型の性質のうち，一様な組織では社会効用が減少し，組織効用が増加するという類

型 6 の割合が多い点，多様な組織では社会効用・組織効用ともに増加する類型 1 と類型 3 の頻度が多い点は Kobayashi et al. (2013)の結論に整合的である．また，社会効用が増加し，組織効用が減少するという類型が一様な組織では類型 5，多様な組織では類型 2 と得られたがこの性質を持つログの存在は Kobayashi et al. (2013)では指摘されておらず，提案手法を適用することではじめて得られたログの性質である．

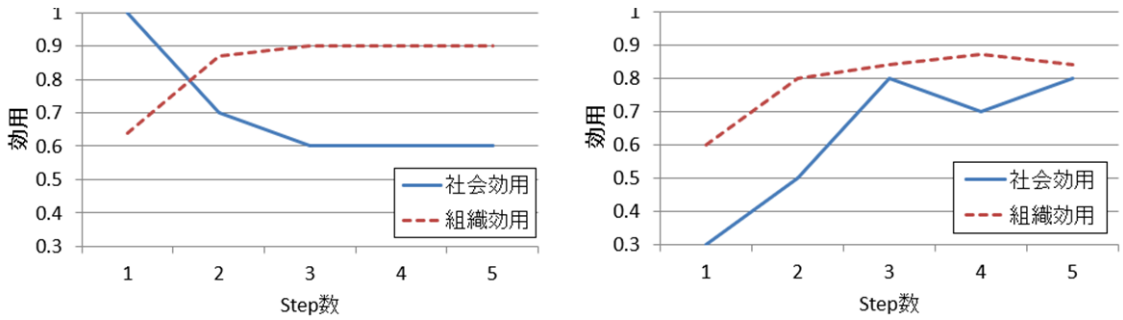


図 4-3 左：逸脱の傾向を示すログ， 右：改善の傾向を示すログ 図に示す 2 つのログは先行研究で見いだされたものであり，それぞれ類型 6 と類型 1 に含まれることを確認した．

一様な組織について，図 4-3 では，類型 6 の性質が，Kobayashi et al. (2013)で扱われているログの性質と一致していることを示している．一様な組織の類型 6 から得られた一つのログを，Kobayashi et al. (2013)で扱われている逸脱のログと比較した．一様な組織・逸脱と先行研究と同様の条件である類型 6 の中に，各 Step おける社会効用，組織効用がすべて同じであるログが存在した．図 4-3 はそのログの社会効用と，組織効用の推移である．クラスターの性質と含まれるログの性質が一致することを確認したとともに，先行研究のログがその性質通りにクラスタリングされていることから，ログのクラスタリングの結果が先行研究に整合的であることを確認した．

同様に，多様な組織についても，先行研究で扱われているログである図 4-3 が類型 1 に含まれていることを確認した．

本節では，改善・逸脱モデルのログをクラスタリングし，得られたクラスターの性質を明らかにすることができた．クラスターの性質のうち多様な組織では改善のログが多く，一様な組織では逸脱のログが多いという点は，先行研究と整合的な結果であった．また，改善にも逸脱にも属さない，社会効用が増加し，組織効用が減少するクラスターは先行研究では未知の性質である．また，クラスターの性質が，含まれるログの性質を説明することを確認できた．先行研究で改善の傾向，逸脱の傾向として扱われていたログが，本研究で得られた同様の性質を示すクラスターにそれぞれ含まれていることを確認した．

図 4-4 は，ログの類型化についての具体的な説明を示している．図 4-4 では多様な組織のパ

ラメータによるログを対象として類型化を行っている．入力として 1000 件あるログのうち図ではランダムに選択された 30 のサンプルの組織効用の推移と社会効用の推移を例として示してる．入力のログをクラスタリングすることにより，3 つの類型に分けていることが確認できる．それぞれの類型ごとにサンプルされたログと類型内の平均を示している．それぞれの類型のサンプルされたログと，類型内の平均から，類型 1 は社会効用，組織効用の推移の両方が単調増加をしていることが分かる．つまり，改善の傾向であるということが分かる．また，この類型には，図 4-3 で示した先行研究で分析されたログと同一のログが含まれている．類型 2 では，社会効用については単調増加であるが，組織効用については一度上昇した後に，減少するという挙動を示している．これは，社会効用と組織効用が上昇するという改善の傾向に当てはまるとは言えない．類型 3 については，類型 1 と同様に推移の両方が単調増加をしており，改善の傾向である．ただし，類型 1 と比較して，両方の時系列の増加が緩やかであることが確認できる．

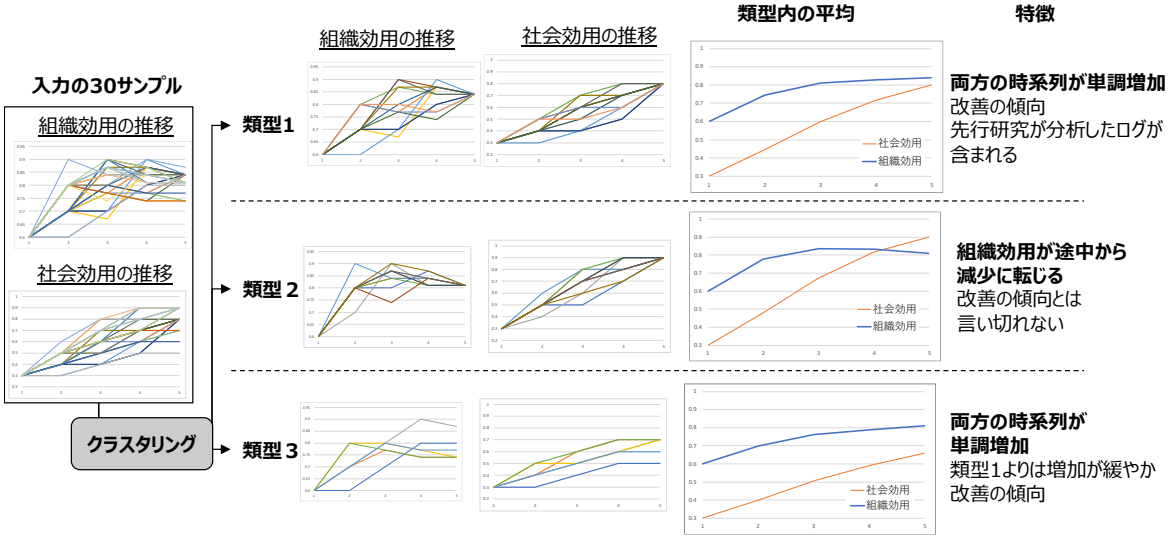


図 4-4 3 つに類型化されたログの実例.

手順２：決定木分析の適用結果

手順２：決定木分析の適用では，先行研究で得られていなかった類型である類型 2 の発生要因を分析するために，類型 1，2 を対象として決定木分析を適用する．

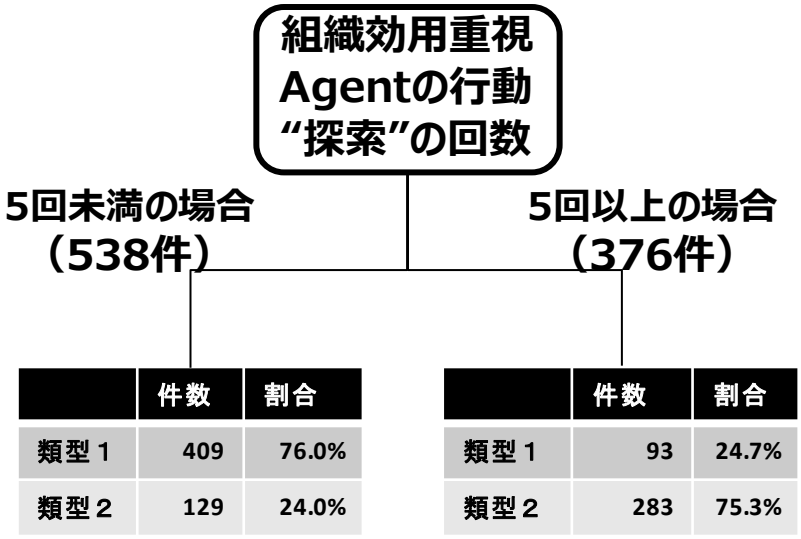


図 4-5 改善・逸脱モデルのクラスター間を分ける決定木から得られたルール.

表 4-6 決定木から得られたルールの学習データでの評価.

	適合率	再現率	F 値	正解率
類型 1	82.1%	74.6%	78.2%	75.5%
類型 2	67.8%	76.7%	72.0%	

図 4-5 では，学習データに決定木分析を行い，得られたルールを示している，類型 1 と類型 2 の学習データを対象とし，要因の候補を Agent ごとの意思決定とし，手順 2：決定木による要因分析を適用した．図 4-5 は"Agent2,3 の探索回数"という要因によって分類すると，5 回未満であるログの中に類型 1 は 76.0 パーセント，5 回以上であるログの中に類型 2 は 75.3 パーセント存在することを示している．さらに，得られたルールを学習データでの評価を行った（表 4-6）．その結果 "Agent2,3 の探索回数"というルールは，類型を 75.5%の正解率で分類できる要因であると分かった．

表 4-7 では，得られたルールの正しさを，個別のログで確認している．類型 1 に含まれるログと，類型 2 に含まれるログの比較を行った．表 4-7 の左右の表はそれぞれ，各 Agent の各 Step における意思決定のログを示している．Agent2，3 の Step5 の意思決定のみが異なるログがそれぞれの類型から発見され，表 4-7 は発見されたログを示している．これによって，探索とい

う要因によって，対象のログの属する類型が変化する，つまりシミュレーション結果が変化することをログレベルで確認できた．

表 4-7 決定木から得られたルールの正しさを確認するため，比較した 2 つのログ：類型 1，類型 2 のそれぞれに含まれるログの比較を行い，Agent2,3 の Step5 の意思決定のみが異なるログが発見された．

類型 1 に含まれるログの 1 つ

Agent	各 Step の意思決定			
	Step2	Step3	Step4	Step5
Agent1	探索	探索	模倣	維持
Agent2	探索	探索	模倣	維持
Agent3	模倣	模倣	模倣	維持

類型 2 に含まれるログの 1 つ

Agent	各 Step の意思決定			
	Step2	Step3	Step4	Step5
Agent1	探索	探索	模倣	維持
Agent2	探索	探索	模倣	探索
Agent3	模倣	模倣	模倣	探索

本節では多様な組織での類型 1 と類型 2 を分ける要因を決定木分析により明らかにした．多様な組織における類型 1 と，類型は”Agent2,3 の探索の意思決定の合計回数が 5 以上かそうでないか”という 1 つのルールで 75.5%の正解率であった．また，得られた要因の正しさを個別のログで確認した．類型 1 に含まれる 1 つのログを取り出し，一部の意思決定が探索である場合，類型 2 に含まれるようになることを確認した．

新たに見出された類型について

適用結果より，社会効用が上昇し，組織効用が減少するという先行研究では扱われていな性質を持つ類型 2 と類型 5 が示された．一方，Kobayashi らの研究においては分布単位に改善・逸脱の分析を行っている．そのため，類型 2 や類型 5 と同様の性質のログは集約されてしまっている可能性がある．しかし，Kobayashi らの研究では，当該の性質に関するに対する言及がない．

類型 2 と類型 5 の性質は逸脱と全く反対のふるまいを示すログである．また，一様な組織で 13.7%，多様な組織で 40.7%と多様度によって頻度が変化している．このことから，多様度の変化によって結果のトレードオフが発生することが示唆される．また，決定木分析より得られた結果は“探索行動”という個人効用の追求が社会効用の上昇だけでなく，組織効用の上昇にも相反する場合があることを示唆している．

4.2.4 実験のまとめ

改善逸脱モデルに提案手法を適用することで，規則性 PSP-AA を抽出すること，既存のエージェントシミュレーションを用いた既存研究での結果と比較し整合的であること，既存研究では得られていない規則性が得られることを確認した．図 4-6 に改善逸脱モデルに提案手法を適用したまとめを記す．

得られた規則性として，「組織効用を重視するエージェントの探索学習の回数が 5 回以上で，組織効用が一度上昇してから下がるという類型になる」という PSP-AA を抽出した．得られた規則性について，既存のエージェントシミュレーションを用いた既存研究での結果と比較し，整合的であることをもって妥当であることを，既存研究では得られていない規則性が得られたことをもって新規性があること確認した．

提案手法によって得られた規則性が既存研究の結果と整合的である点は次である．既存研究での知見として，組織の構成員の多様性が改善と逸脱を分ける要因であることに対して，本研究での結果は，多様性の高い組織では改善の類型の割合が多く，多様性の低い組織では逸脱の類型の割合が多いという結果であった．さらに，既存研究で，多様性の高い組織の実行結果から取り出した改善のログを分析しているのに対して，本研究では先行研究で扱われているログが本研究での改善の類型の中に含まれていたことを確認した．既存研究では示していない規則性を得れる場合がある点について，一度増えてから減るという，時間変化を扱っているからこその類型が得られた．

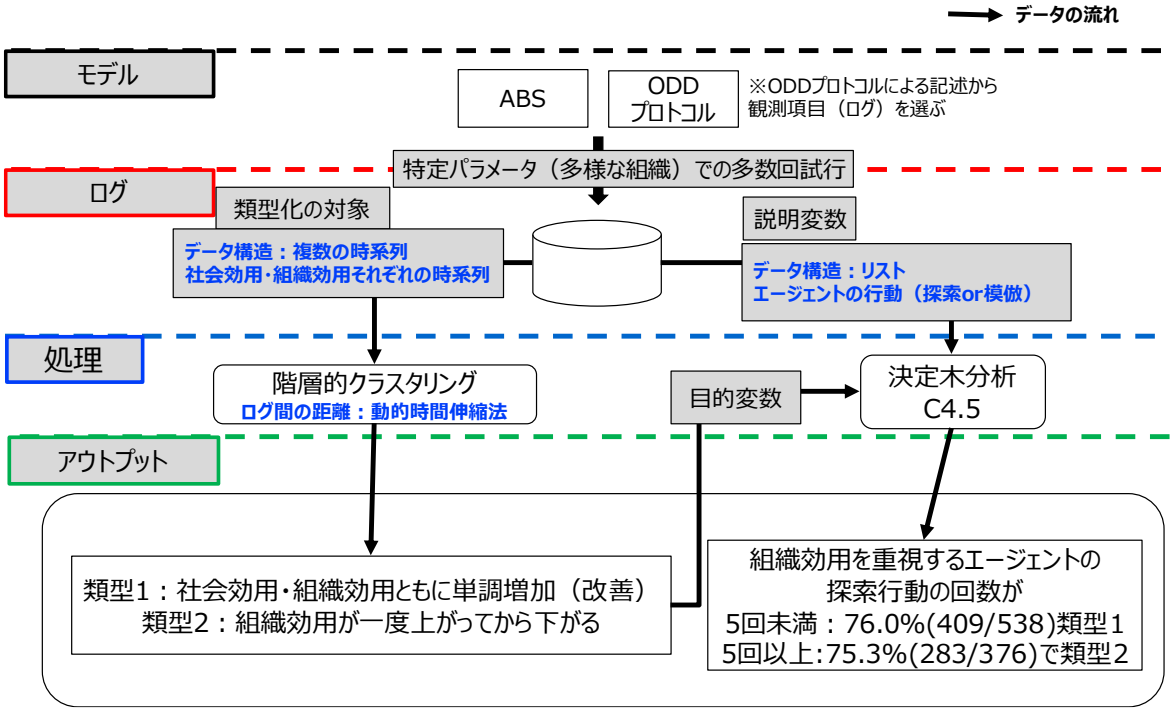


図 4-6 改善・逸脱モデルへの適用のまとめ．

4.3 適用 2：金融シミュレーションへの適用

本節においては，金融システムにおける破綻の伝播を表現する既存のシミュレーションモデル(菊地ら 2016)を対象とし，提案手法を適用する．まず，当該の破綻伝播モデルについて説明をし，その後手法を適用する．

4.3.1 先行研究：破綻伝播モデル

菊地ら(2016)はリーマンショックや欧州危機等，金融機関の破綻が金融システム全体に波及するリスク(Allen and Carletti 2010) (Kaufman 2000)，いわゆる“システミック・リスク”(May and Arinaminpathy 2010)に注目し，金融機関を取り巻く金融規制・運営制約や中央銀行の政策が, 金融システムの安定性に与える影響の分析をエージェントシミュレーションにより行った．

上記の目的を達成するために，菊地らは以下の問題にアプローチしている．(1)市場性資産の価格下落により主要金融機関における破綻の連鎖は生起し得ないのか．(2)金融規制・運営制約により破綻の連鎖のリスク（破綻数など）は低減されるか．(3)中央銀行の政策によって破綻の連鎖のリスク（破綻数など）はどこに転嫁するか．方法論としては，共通資産への価格ショックを織り込んだ破綻伝播と資金流動性による破綻を陽に表現し，金融規制・運営制約や中央銀行の政策を扱えるエージェントモデルを構築している．

主要な結果としては，(1)市場性資産の価格変動による金融機関の財務・信用状況の変化を通じた破綻の連鎖は生起し得ること．(2)金融規制・運営制約の組み合わせによっては，むしろ個別金融機関の破綻可能性を高めること．(3)中央銀行の政策は，先行研究で言及されているものとは別のリスクをもたらす可能性があること，を示している

4.3.2 実験の設定

本節では，破綻伝播モデルに本手法を適用する．具体的に，ODD プロトコルの記述に従ったクラスタリングを行うことで，クラスターの性質を明らかにし，シミュレーション結果の分類を行う．

手順 1：クラスタリングの適用の設定

表 4-8 利用したデータとアルゴリズム

データ 1 ：ログの生成条件	Agent0 の市場性資産と現金の保有割合を変化させ，それぞれ 500 試行ずつのログ集合．それ以外のパラメータは固定（菊池ら 2016）と同一のパラメータ. Agent0 は大口の資金供給先である．
データ 2 ：“モデルが扱っている現象の発生プロセス”のログ	破綻エージェントの順番と，破綻理由を文字列として扱う．
アルゴリズム	階層的クラスタリング ward 法 ログ間の距離：レーベンシュタイン距離 クラスター数：CH 基準により決定．

本節では，破綻伝播モデルにログのクラスタリングを適用し，クラスターの性質を明らかにできることを示す．具体的には，得られたクラスターの性質を明らかにし，各クラスターに含まれるログの分析を行う．

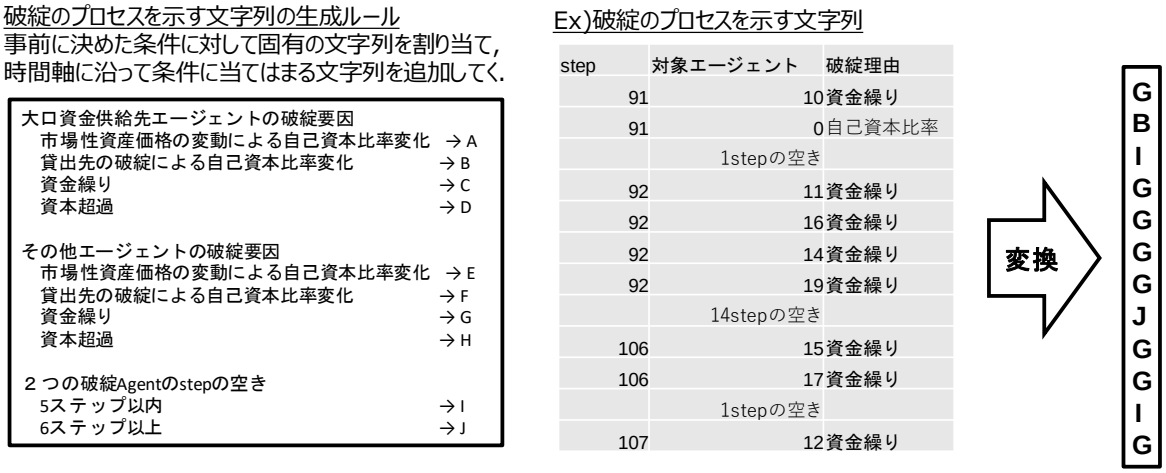


図 4-7 ログの変換ルールと例

本研究でログのクラスタリングを適用した際，用いたデータとアルゴリズムの詳細を説明する．また，得られた結果を説明する．

ログのクラスタリングを適用した際の，データとアルゴリズムの概要を表 4-8 に示す．（データ 1：ログの生成条件）特定の Agent の市場性資産と現金の保有割合を 10%ずつ変化させ，それぞれ 500 試行ずつのログの集合．ほかのパラメータは固定している．(データ 2：“モデルが扱っている現象の発生プロセス”のログ)ODD プロトコルで記述されたモデルの記述から得られた観測項目と性質を利用している．観測項目：破綻エージェントの順番と，破綻理由を文字列に変換し扱う．ログの性質：出現頻度，破綻 step 数，保有市場性資産(アルゴリズム）階層クラスタリングの ward 法を利用．ログ間の距離はレーベンシュタイン距離を利用した．

本研究では大口の資金供給先である特定の Agent を中心とした破綻に着目し，シミュレーション結果を分類する．特定の Agent（以下 Agent0）の市場性資産と現金の保有割合を 10%ずつ変化させ，それぞれ 500 試行ずつのログの集合．ほかのパラメータは同様に生成したものを固定している．

破綻伝播モデルの ODD プロトコルによる記述から得られた観測項目と性質を利用している．観測項目：破綻エージェントの順番と，破綻理由を文字列に変換し扱う．図 4-7 は“エージェントの行動”のログの例と，文字列への変換ルールである．ログの性質は，出現頻度，破綻 step 数，保有市場性資産で示す．クラスタリング手法は，階層クラスタリング手法の ward 法を利用しており，ログ間の距離は時系列データであるため，文字列データの距離を求める手法であるレーベンシュタイン距離を利用している．

手順２：決定木分析の適用の設定

手順２：決定木分析の適用した際の，データとアルゴリズムの概要を，表 4-9 に示す．（データ 1）手順１で生成したログと同様である 5500 件のログを対象としている．（データ 2）“エージェントの行動”のログは，Agent・Step ごとの市場性資産の購入量と，資金繰りの量である．（データ 3）それぞれのログがどの類型に属すかは手順１で得られたログごとの属す類型である．（アルゴリズム）決定木による要因分析のアルゴリズムは，C4.5 を利用している．

表 4-9 利用したデータとアルゴリズム（連鎖破綻モデルへの適用の手順 2）

データ１：ログの生成条件	手順１と同様である 5500 件のログ．
データ２：“エージェントの行動”のログ	Agent・Step ごとの市場性資産の購入量と，資金繰りの量．
データ３：それぞれのログがどの類型に属すか	手順１で得られたログごとのラベル．
アルゴリズム	C4.5

4.3.3 実験の結果

すべてのログに対して手順 1 を適用した結果（図 4-8），CH 基準によりクラスターを 3 つに分け（図 4-9），各クラスターの性質が得られた(表 4-10).

図 4-8 は適用結果の分布を示している．類型 1 は僅差ではあるが最も出現頻度が低い． 3 つのクラスターのうち，類型 1 の保有する Agent0 の市場性資産は最も少なく，Agent0 の破綻ステップ数は最も遅い. 類型 2 は類型 1 と似た性質のクラスターである. 類型 1 と似てはいるが，比較し Agent0 の保有する市場性資産が多く，Agent0 の破綻ステップ数が早い．類型 3 はほかの 2 つと性質が大きく異なり，さらに出現頻度も最も多い．Agent0 の保有する市場性資産は最も多く，Agent0 の破綻ステップ数は最も早い.

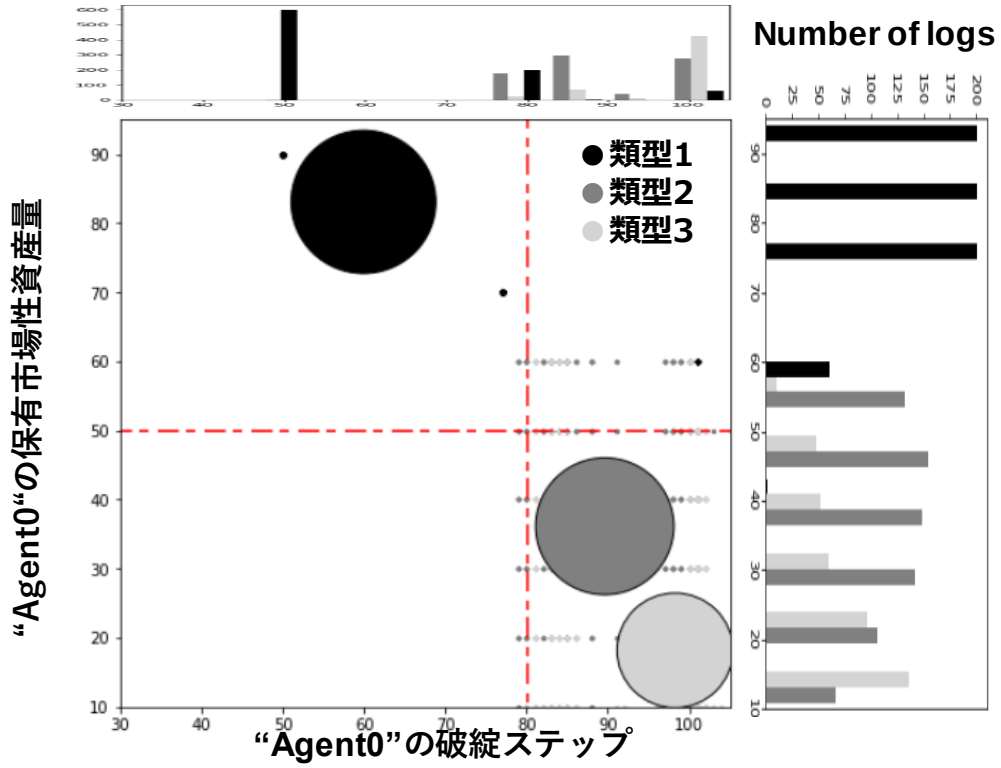
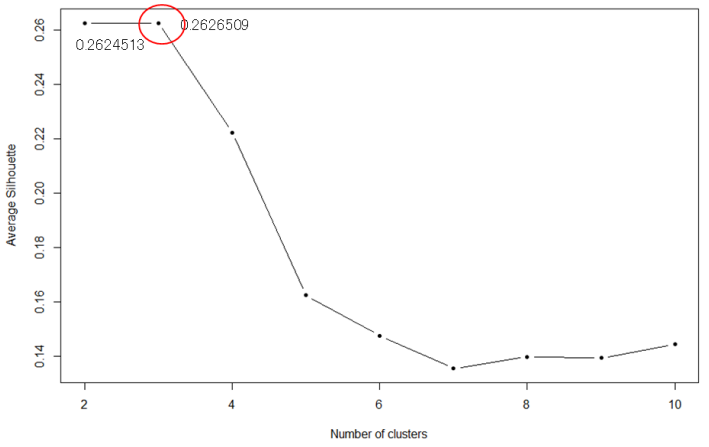


図 4-8 手法を適用し得られた類型

表 4-10 手法により得られた類型の性質

	出現頻度	破綻 step 数	保有市場性資産
類型 1	30.9%	96.4	22.6
類型 2	31.1%	91.8	36.8
類型 3	38.0%	60.0	83.1



Sihouetteは、他のクラスタと自身のクラスタがどれほど似ているかの尺度
任意の距離基準によるクラスタリングに適用可能

図 4-9 CH 基準によるクラスター数の決定

3 つのログクラスターから分かる傾向として，保有する市場性資産の量が多いほど，破綻ステップ数が早まる傾向にある．モデルでは市場性資産の価格は減少していく設定になっており，モデルの前提と整合的な結果である．

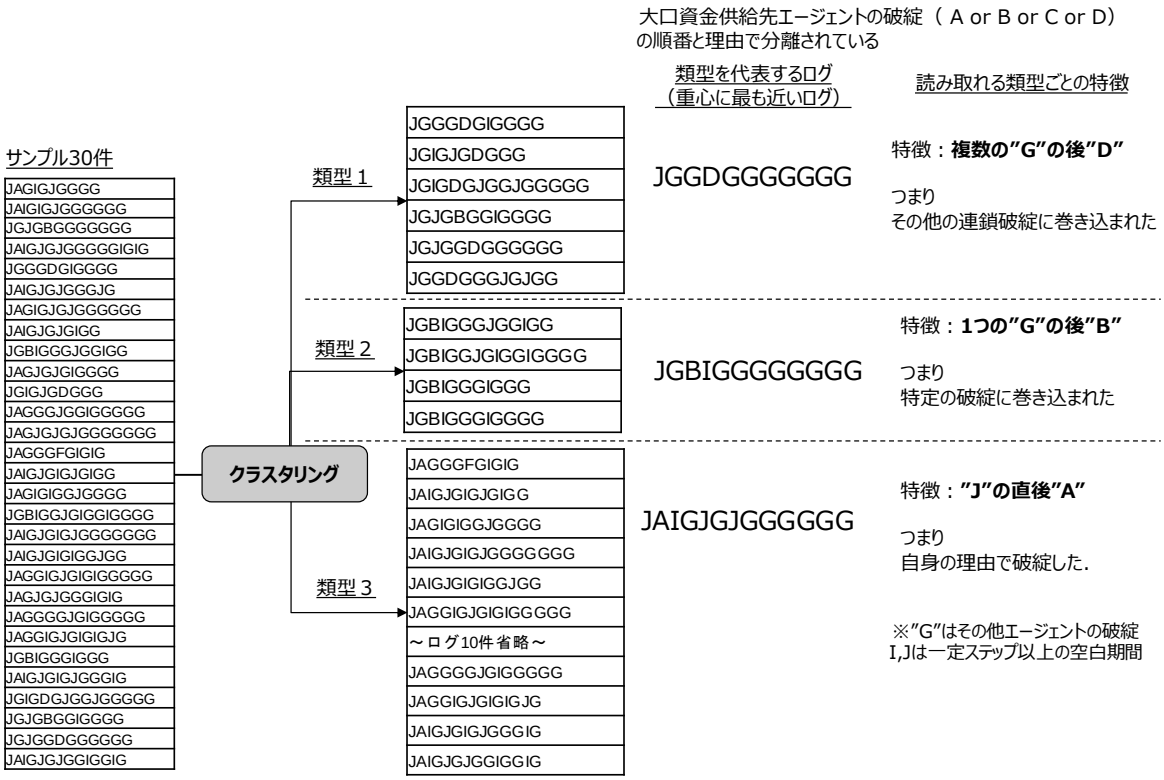


図 4-10 3 つに類型化されたログの例.

どのような特徴がクラスタリングによって分けられたかを図 4-10 で説明する．図 4-10 左は全ログから 30 ログをランダムに抽出したものを示している．この状態ではどのように分離されるかは自明ではない．図 4-10 ではクラスタリングを行うことで，3 つの類型に分かれることを示してる．その結果の実際のログと，クラスターの重心に最も近いクラスターを代表するログを示している．これらの結果から，類型 1 は複数の“G”の後に“D”がくるという特徴で分離されていることが分かる．また，類型 2 は 1 つの“G”の後，“B”がくるという特徴で分離されていることが分かる．さらに，類型 3 は“J”の直後“A”がくるという特徴で分離されていることが分かる．それぞれの文字を意味すること（図 4-7）を解釈すると，類型 1 は Agent0 以外の Agent が引き起こした連鎖破綻に巻き込まれて破綻した．類型 2 は 1 体のエージェントの破綻に巻き込まれた．類型 3 は Agent0 が単独で破綻した．

図 4-11 では得られた類型を代表するログを説明している．ログはそれぞれのクラスターの重心に最も近いログから抽出してる．類型 1 では，度重なる 2 社の破綻を受け止めたものの，3 社目の破綻で耐えきれず，それまで金融システムを支えていた Agent0 が破綻したことで，7 社の連鎖的破綻が生じた．類型 2 では，1 社の破綻を受け，大口資金供給先だった Agent0 も影響を受け，連鎖破綻となったもの．その後の市場性資産下落を受け，更に破綻総数が増加した．類型 3 は，Agent0 が最初に単独で破綻したケース．その後の市場性資産価格の下落を受け，断続的に破綻が生じた．最終的には大口資金供給先である Agent0 の不在も響き，7 社の連鎖破綻となった．

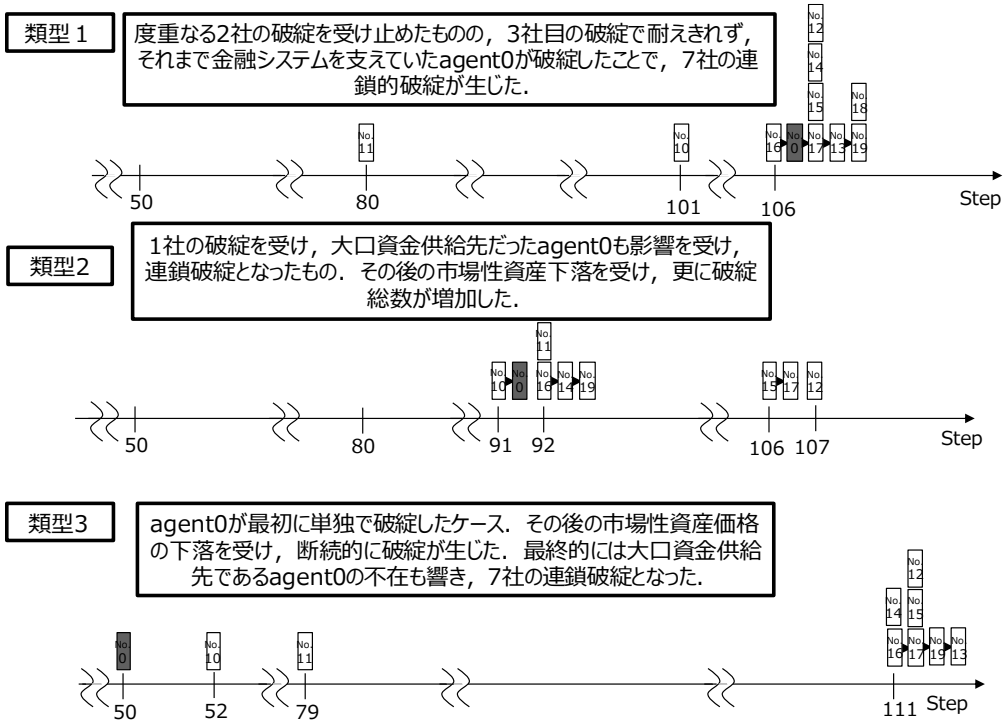


図 4-11 各類型の代表的なログの分析

図 4-12,13,14 では図 4-11 で示したそれぞれのログの Agent0 の自己資本比率，資本の推移を示している．図 4-12 では類型 1 のログについてであり，ステップ 80 付近で Agent11 が破綻したことによって，資本が大きく減少しているがまだ破綻はしておらず，その後続いて Agent10 の破綻までは耐えていたが，Agent16 の破綻により債務超過となってしまい破綻が起こっている．図 4-13 では 90 ステップ付近で，Agent10 が破綻したことにより，資本が減少し，自己資本比率が，最低自己資本比率である 1%を割ってしまうことにより破綻が起こっている．図 4-14 では，他のエージェントの破綻の影響は受けていないが，保有している市場性資産の資産価格が減少し，50 ステップ付近で単独で破綻している．

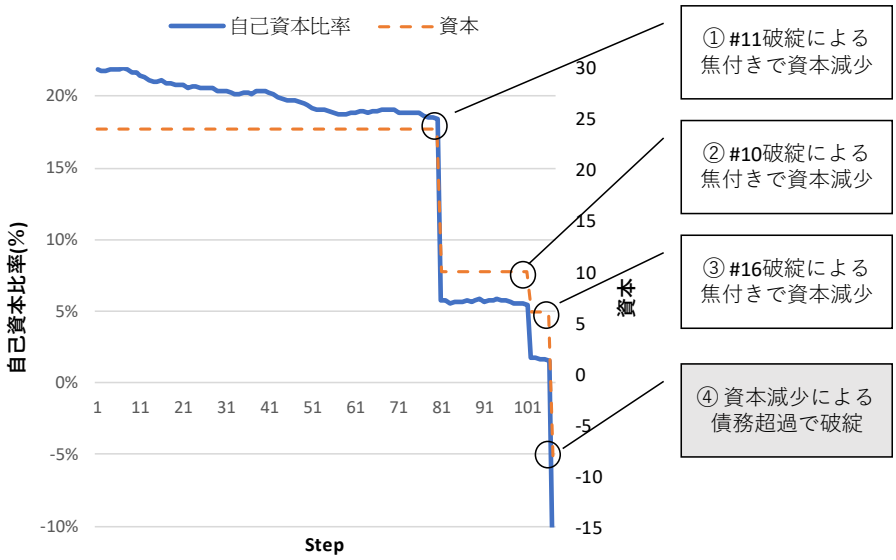


図 4-12 類型 1 での Agent0 が破綻に至る過程.

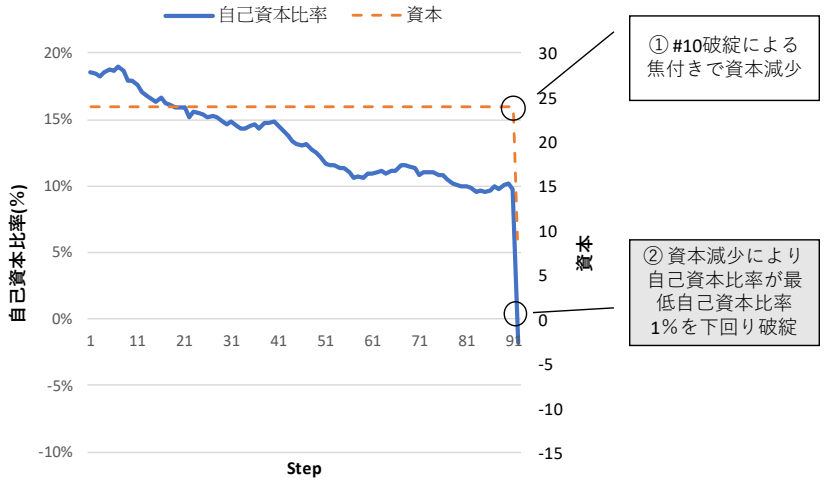


図 4-13 類型 2 での Agent0 が破綻に至る過程.

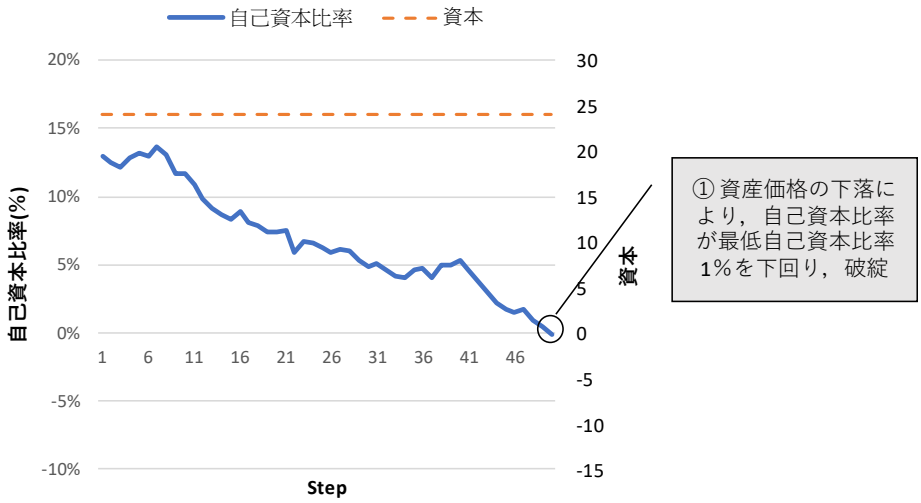


図 4-14 類型 3 での Agent0 が破綻に至る過程.

クラスターごとのログを集計した値での性質だけではわからない詳細の性質を明らかにするため，各クラスターから，クラスターの中心に近いログから代表的なログを抽出し，ログ分析を行った.

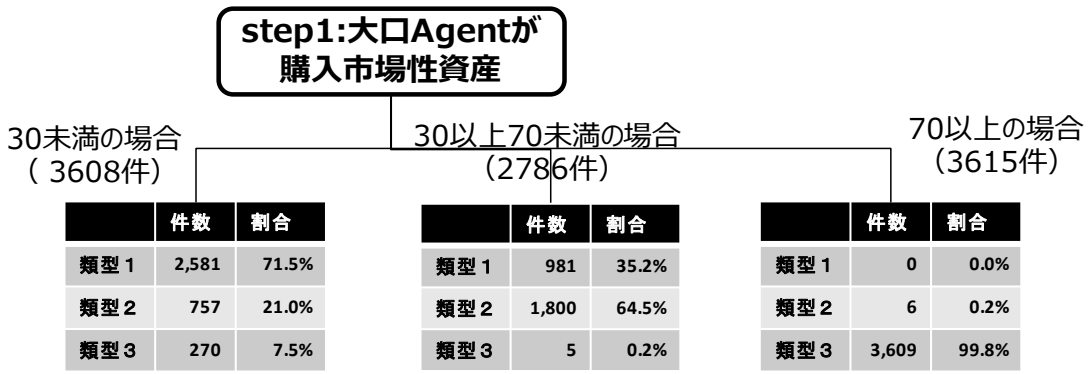
類型 1 は度重なる 2 社の破綻を受け止めたものの，3 社目の破綻で耐えきれず，それまで金融システムを支えていた Agent0 が破綻したことで，7 社の連鎖的破綻が生じた．類型 2 は 1 社の破綻を受け，大口資金供給先だった Agent0 も影響を受け，連鎖破綻となったもの．その後の市場性資産下落を受け，　更に破綻総数が増加した．類型 3 は Agent0 が最初に単独で破綻したケース．その後の市場性資産価格の下落を受け，断続的に破綻が生じた．最終的には大口資金供給先である Agent0 の不在も響き，7 社の連鎖破綻となった.

類型 1 , 2 は集計した性質では似ていたが，ログ分析を行うと異なる解釈が可能なログであることが分かる． 3 つのログは Agent0 の破綻 step の早さが類型 1 から順に遅くなっており， 3 つの類型全体での傾向と同様である.

手順２：決定木分析の適用結果

手順２:決定木分析の適用では,手順１で得られた３つの類型の発生要因を分析するために,決定木分析を適用する.

図 4-15 では，学習データに決定木分析を行い，得られたルールを示している，類型１と類型２の学習データを対象とし，要因の候補を Agent ごとの意思決定とし，手順２：決定木による要因分析を適用した．図 4-15 は Step1 の Agent0 の購入市場性資産量によって類型をどれくらい分離できるかを示している．Agent0 の購入市場性資産量が 30 未満の場合は，71.5％（2581/3608 件）が類型１に，Agent0 の購入市場性資産量が 30 以上 70 未満の場合は，64.5％（1800/2786 件)が類型２に,Agent0 の購入市場性資産量が 70 以上の場合は,99.8％(3609/3615 件）が類型３になることが読み取れる.



	件数	割合
類型１	2,581	71.5%
類型２	757	21.0%
類型３	270	7.5%

	件数	割合
類型１	981	35.2%
類型２	1,800	64.5%
類型３	5	0.2%

	件数	割合
類型１	0	0.0%
類型２	6	0.2%
類型３	3,609	99.8%

図 4-15 破綻伝播モデルのクラスター間を分ける決定木.

4.3.4 実験のまとめ

連鎖破綻モデルに提案手法を適用することで，規則性 PSP-AA を抽出すること，既存のエージェントシミュレーションを用いた既存研究での結果と比較し整合的であること，既存研究では得られていない規則性が得られることを確認した．図 4-16 に適用のまとめを示す．

得られた規則性として,「大口資金供給先であるエージェントの購入する市場性資産量によって連鎖破綻の起こり方が 3 つに分かれる」という PSP-AA が得られた．

提案手法によって得られた規則性が既存研究の結果と整合的である点は，既存研究では，連鎖破綻は起こり得ることを示すにとどまっていたのに対して，本研究では連鎖破綻についての 3 つの類型を示した．

既存研究では示していない規則性を得れる場合があるという点については，存在を示すにとどまっていた連鎖破綻についての次の 3 つの典型的な類型を示した．(1)複数の破綻に耐え切れず破綻する．(2)他社に巻き込まれて破綻が発生する．(3)単独で破綻し，他に破綻伝播していく．

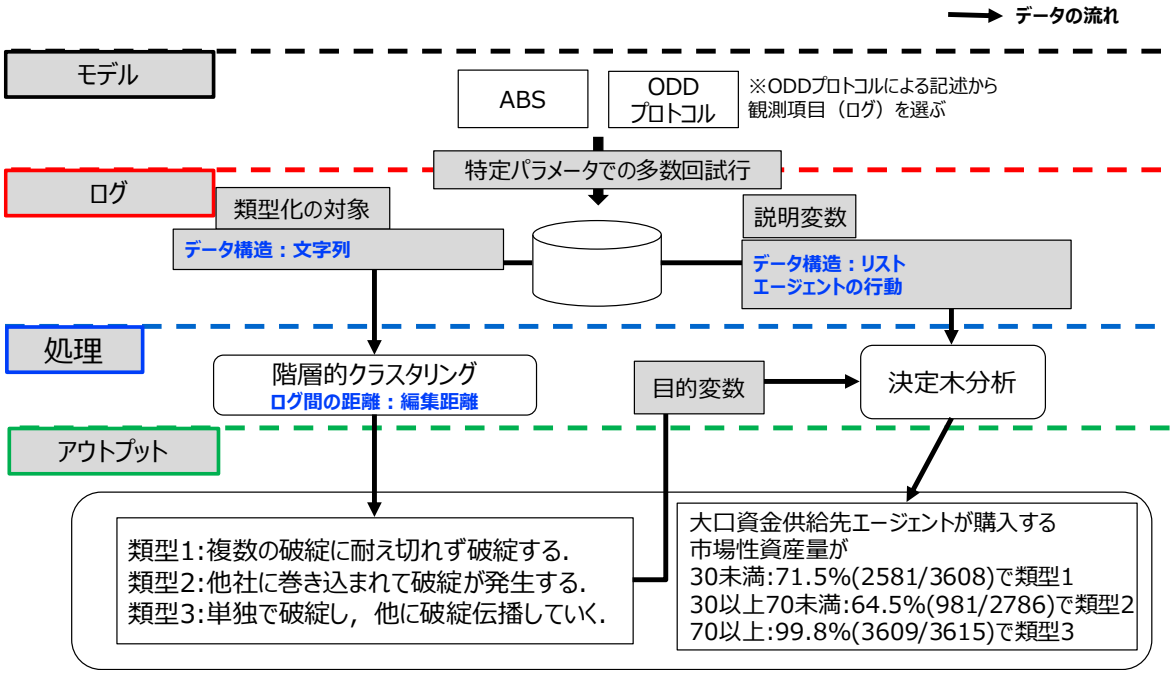


図 4-16 連鎖破綻モデルへの適用のまとめ

第 5 章　貢献と課題

5.1 本研究のまとめ

本研究では，従来から重要性は指摘されていたが解決方法が十分に示されていないエージェントシミュレーションにおけるプロセスの分析方法について，PSP-AA という論文中で定義された”シミュレーションのプロセスについての類型と，その類型と関連があるエージェントの行動”についての規則性，つまりシミュレーションの結果に大きな影響を与える特定のエージェントの行動が存在するということを，異なるシミュレーションモデルにおいて示し，同一の手法で抽出可能であることを分居モデル，改善逸脱モデル，連鎖破綻モデルという複数のエージェントシミュレーションで実証した．

提案手法を分居モデルに適用し，特定の初期状態からのログについて，特定のエージェントの行動によって，特徴の異なる 2 つの類型に分けることができる例を示した．

また，改善逸脱モデルに提案手法を適用することで，規則性 PSP-AA を抽出すること，既存のエージェントシミュレーションを用いた既存研究での結果と比較し整合的であること，既存研究では得られていない規則性が得られることを確認した．

同様に，連鎖破綻モデルに提案手法を適用することで，規則性 PSP-AA を抽出すること，既存のエージェントシミュレーションを用いた既存研究での結果と比較し整合的であること，既存研究では得られていない規則性が得られることを確認した．

上記の様に PSP-AA の抽出を実証できたことはエージェントシミュレーションにおけるプロセスの分析を改善できる可能性を示している．

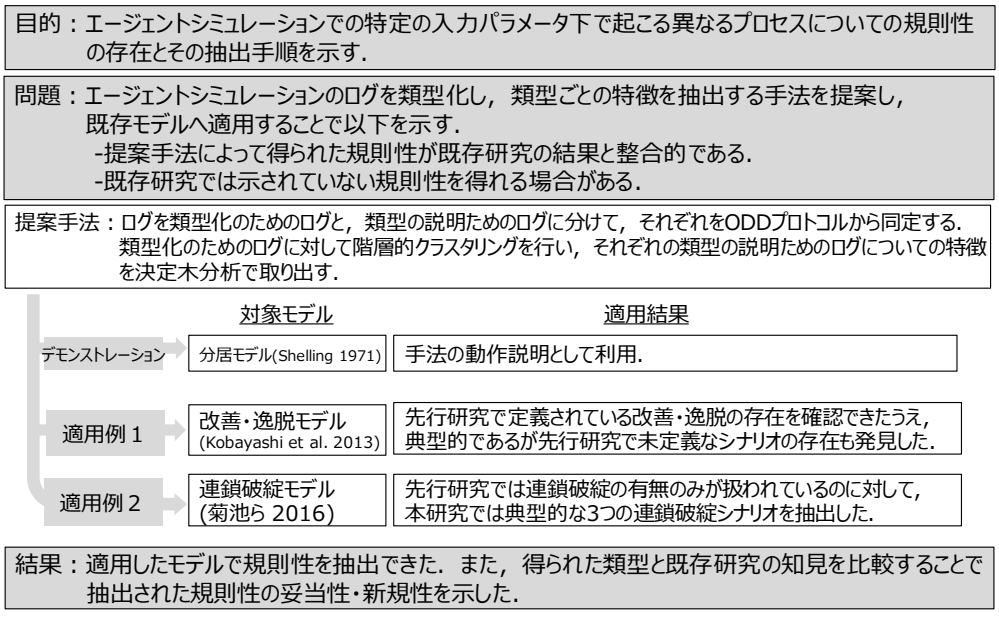


図 5-1 全体のまとめ

5.2 今後の課題

今後の課題は以下の 3 点である．1.あらゆるエージェントシミュレーションへの適用の検証. 2.アルゴリズムの計算時間．3.PSP-AA の応用研究.

本研究の提案手法は，他のモデルへの適用の限界があり得ると考えられる．つまり，本研究の提案手法を用いることですべてのエージェントシミュレーションについて，本研究で定義している規則性(PSP-AA)が抽出できるとは限らない．それは今回用いたアルゴリズムが対象のモデルで抽出すべき特徴を捉えられることを断定できないためである．

類型化はログ間の距離の定義に依存するため，本研究で用いた距離計算のアルゴリズム（動的時間伸縮法，レーベンシュタイン距離）で分離できない特徴を対象のモデルの挙動が示す場合が考えらえる．決定木分析の説明変数がステップ・エージェントごとの行動であり，時間遅れやエージェントの入れ替わりなどに意味がない場合も違いデータとして扱ってしまう．これらは用いるアルゴリズムの交換で対応できる可能性はあるが本研究のスコープ外である．

また，アルゴリズムの計算時間も課題である．本研究で利用しているアルゴリズム（動的時間伸縮法，階層的クラスタリング ward 法，レーベンシュタイン距離，C4.5）はデータ量の増加に対して，線形時間で解けない．そのため，大規模データへの適用に限界があり，効率化が求められる．

PSP-AA を用いた次のような応用が考えられる．1)ゲーミングとの接地．ゲーミングにおいて，ケースの洗い出しと，重要な意思決定を抽出可能であると考えられる．それにより，ファシリテーションのサポートを可能にできる可能性がある. 2)シミュレーションからの物語生成. 典型的な振る舞いと，その分岐点を自然言語化することでシミュレーションからの物語生成する．手法としては，ログを文字列として扱っているため，自然言語処理で用いられる様々な手法(Sebastiani 2002) (Bird, Klein and Loper 2009)が適用できる可能性がある．3)政策決定のための根拠．特定のパラメータを，現実世界の現在の状態にすることで，これからどのようなことが起こりえるか．また，起こることがどのような要因によって引き起こされるかをモデル上で俯瞰することが可能となる. これは, 不確実性のある状況下での意思決定(Polasky et al. 2011)に有用であると考えられる.

参考文献

1. Aghabozorgi S., Shirkhorshidi A. I., & Wah T. Y. (2015). Time-series clustering - A decade review. Information Systems, 53, 16–38.

2. Allen F., & Carletti E. (2010). An Overview of the Crisis: Causes, Consequences, and Solutions. International Review of Finance, 10(1), 1–26.

3. Axelrod R. (1997). The complexity of cooporation, Agent-Based Model of Competition and Collaboration. Princeton University Press.

4. Bazzan A. L., & Franziska K. (2014). A review on agent-based technology for traffic and transportation. The Knowledge Engineering Review, 29(03), 375–403.

5. Benoit C., & Hutzler G. (2005). Automatic tuning of agent-based models using genetic algorithms. International Workshop on Multi-Agent Systems and Agent-Based Simulation. Springer, Berlin, Heidelberg, 41–57.

6. Berndt D. J., & Clifford J. (1994). Using dynamic time warping to find patterns in time series. KDD Workshop, 10(16), 359–370.

7. Bird S., Klein E., & Loper E. (2009). Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc.

8. Caliński T., & Harabasz J. (1974). A Dendrite Method For Cluster Analysis. Communications in Statistics, 3(1), 1–27.

9. Crooks A. T. (2010). Constructing and implementing an agent-based model of residential segregation through vector GIS. International Journal of Geographical Information Science, 24(5), 661–675.

10. Domic N. G., Goles E., & Rica S. (2011). Dynamics and complexity of the Schelling segregation model. Physical Review E - Statistical, Nonlinear, and Soft Matter Physics, 83(5), 1–13.

11. Everitt B. S., Landau S., Leese M., & Stahl D. (2011). Cluster Analysis (5 edition). Wiley.

12. Fachada N., Lopes V. V., Martins R. C., & Rosa A. C. (2017). Model-independent comparison of simulation output. Simulation Modelling Practice and Theory, 72, 131–149.

13. Fayyad U., Piatetsky-Shapiro G., & Smyth P. (1996). From Data Mining to Knowledge Discovery in Databases. AI Magazine, 17(3), 37–54.

14. Gauvin L., Vannimenus J., & Nadal J. P. (2009). Phase diagram of a Schelling segregation model. European Physical Journal B, 70(2), 293–304.

15. Grimm V., Berger U., Bastiansen F., Eliassen S., Ginot V., Giske J., … DeAngelis D. L. (2006). A standard protocol for describing individual-based and agent-based models. Ecological Modelling, 198(1–2), 115–126.

16. Grimm V., Berger U., DeAngelis D. L., Polhill J. G., Giske J., & Railsback S. F. (2010). The ODD protocol: A review and first update. Ecological Mode,Lling, 221(23), 2760–2768.

17. Grimm V., Revilla E., Berger U., Jeltsch F., Mooij W. M., Railsback S. F., … DeAngelis D. L. (2005). Pattern-Oriented Modeling of Agent-Based Complex Systems: Lessons from Ecology. Science, 310(5750), 987–991.

18. Gusfield Dan. (1997). Algorithms on strings, trees and sequences: computer science and computational biology. Cambridge University Press.

19. Happe K. (2005). Agent-based modelling and sensitivity analysis by experimental design and metamodelling: an application to modelling regional structural change. International Congress of the European Association of Agricultural Economists.

20. Hartig F., Calabrese J. M., Reineking B., Wiegand T., & Huth A. (2011). Statistical inference for stochastic simulation models - theory and application. Ecology Letters, 14(8), 816–827.

21. Hintze Jerry L, & Nelson Ray D. (1998). Violin Plots: A Box Plot-Density Trace Synergism, 52(2), 181–184.

22. Imai M. (1986). Kaizen: The Key To Japan's Competitive Success. McGraw-Hill/Irwin.

23. Jeong Y S, Jeong Myong K, & Omitaomu Olufemi A. (2011). Weighted dynamic time warping for time series classification. Pattern Recognition, 44(9), 2231–2240.

24. Kahl C H, & Hansen H. (2015). Simulating Creativity from a Systems Perspective : CRESY. Journal of Artificial Societies and Social Simulation, 18(1), 4.

25. Kaufman G. G. (2000). Banking and currency crises and systemic risk: Lessons from recent events. Economic Perspectives, 24(3), 9–28.

26. Kobayashi T., Takahashi S., Kunigami M., Yoshikawa A., & Terano T. (2013). Is There Innovation or Deviation ? Analyzing Emergent Organizational Behaviors through an Agent Based Model and a Case Design. The Fifth International Conference on Information, Process, and Knowledge Management EKNOW 2013, 166–171.

27. Kohavi R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. IJCAI'95 Proceedings of the 14th International Joint Conference on Artificial Intelligence, 2, 1137–1143.

28. Lee J. S., Filatova T., Ligmann-Zielinska A., Hassani-Mahmooei B., Stonedahl F., Lorscheid I., … Parker, D. C. (2015). The complexities of agent-based modeling output analysis. Journal of Artificial Societies and Social Simulation Simulation, 18(4), 1–26.

29. Liao T. W. (2005). Clustering of time series data - A survey. Pattern Recognition, 38(11), 1857–1874.

30. Lorscheid I., Heine B. O., & Meyer M. (2012). Opening the "Black Box" of Simulations: Increased Transparency and Effective Communication Through the Systematic Design of Experiments. Computational and Mathematical Organization Theory, 18(1), 22–62.

31. Marshall B. D., & Galea S. (2014). Formalizing the role of agent-based modeling in causal inference and epidemiology. American Journal of Epidemiology, 181(2), 92–99.

32. Martínez I., Wiegand T., Camarero J. J., Batllori E., & Gutiérrez E. (2011). Disentangling the Formation of Contrasting Tree-Line Physiognomies Combining Model Selection and Bayesian Parameterization for Simulation Models. The American Naturalist, 177(5), E136–E152.

33. May F., Giladi I., Ristow M., Ziv Y., & Jeltsch F. (2013). Metacommunity, mainland-island system or island communities? Assessing the regional dynamics of plant communities in a fragmented landscape. Ecography, 36(7), 842–853.

34. May R. M., & Arinaminpathy N. (2010). Systemic risk: the dynamics of model banking systems. Journal Of The Royal Society Interface, 7(46), 823–838.

35. Molina J.A.E., Clapp C. E., Linden D. R., Allmaras R. R., Layese M. F., Dowdy R. H., & Cheng H. H. (2001). Modeling the incorporation of corn (Zea mays L.) carbon from roots and rhizodeposition into soil organic matter. Soil Biology and Biochemistry, 33(1), 83–92.

36. Müller B., Bohn F., Dreßler G., Groeneveld J., Klassert C., Martin R., … Schwarz N. (2013). Describing human decisions in agent-based models - ODD+D, an extension of the ODD protocol. Environmental Modelling and Software, 48, 37–48.

37. Müller B, Balbi S, Buchmann C. M., de Sousa L., Dressler G., Groeneveld J., … Weise H. (2014). Standardised and transparent model descriptions for agent-based models: Current status and prospects. Environmental Modelling and Software, 55, 156–163.

38. Murray-Rust D., Robinson D. T., Guillem E., Karali E., & Rounsevell M. (2014). An open framework for agent based modelling of agricultural land use change. Environmental Modelling and Software, 61, 19–38.

39. Murtagh F., & Contreras P. (2012). Algorithms for hierarchical clustering: An overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2(1), 86–97.

40. Ohori K., Kobayashi N., Obata A., Takahashi A., & Takahashi S. (2012). Decision support for management of agents' knowledge and skills with job rotation in service-oriented organization. Proceedings of the Annual Hawaii International Conference on System Sciences, 1492–1501.

41. Patel B. R., & Rana K. K. (2014). A Survey on Decision Tree Algorithm For Classification. Ijedr, 2(1), 1–5.

42. Polasky S., Carpenter S. R., Folke C., & Keeler B. (2011). Decision-making under great uncertainty: Environmental management in an era of global change. Trends in Ecology and Evolution, 26(8), 398–404.

43. Quinlan J. R. (1993). C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc.

44. Quinlan J. R. (1996). Improved use of continuous attributes in C4. 5. Journal of Artificial Intelligence Research, 4, 77–90.

45. Railsback S. F., & Grimm V. (2011). Agent-Based and Individual-Based Modeling: A Practical Introduction. Princeton Univ Pr.

46. Rajapakse C., & Terano T. (2013). An Agent-based Model to Study the Evolution of Service Systems through the Service Life Cycle. Agent-Based Approaches in Economic and Social Complex Systems VIII: Post-Proceedings of The AESCS International Workshop 2013, 177–188.

47. Russell S., & Norvig P. (2016). Artificial Intelligence A Modern Approach (Global Edi). Pearson Education Limited.

48. Schelling T. C. (1971). Dynamic models of segregation. The Journal of Mathematical Sociology, 1(2), 143–186.

49. Sebastiani F. (2002). Machine Learning in Automated Text Categorization. Journal ACM Computing Surveys (CSUR), 34(1), 1–47.

50. Senin P. (2008). Dynamic Time Warping Algorithm Review. Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA, 855, 1–23.

51. Singh A., Vainchtein D., & Weiss H. (2009). Schelling's segregation model: Parameters, scaling, and aggregation. Demographic Research, 21, 341–366.

52. Sobol I. M. (1993). Sensitivity analysis for nonlinear mathematical models. Math. Model. Computer.Exp, 1(4), 407–414.

53. Szimanski F., Ralha C. G., Wagner G., & Ferreira D. R. (2013). Improving Business Process Models with Agent-based Simulation and Process Mining. BPMDS 2013, EMMSAD 2013: Enterprise, Business-Process and Information Systems Modeling, 124–138.

54. Terano T., Kobayashi T., Takahashi S., Kunigami M., & Yoshikawa A. (2013). Case Meets Agent-Simulation : Toward Model-Based Case Development in a Service Domain. Proc. 2013 Frontier in Service Conference.

55. Thiele J. C., Winfried K., & Grimm V. (2014). Facilitating Parameter Estimation and Sensitivity Analysis of Agent-Based Models: A Cookbook Using NetLogo and R. Journal of Artificial Societies and Social Simulation, 17(3), 11.

56. Thomassey S., & Fiordaliso A. (2006). A hybrid sales forecasting system based on clustering and decision trees. Decision Support Systems, 42(1), 408–421.

57. Topping C. J., Høye T. T., & Olesen C. R. (2010). Opening the black box-Development, testing and documentation of a mechanistically rich agent-based model. Ecological Modelling, 221(2), 245

58. Vinković D., & Kirman A. (2006). A physical analogue of the Schelling model. Proceedings of the National Academy of Sciences, 103(51), 19261–19265.

59. Yamashita T., Matsushima H., & Noda I. (2014). Exhaustive analysis with a pedestrian simulation environment for assistant of evacuation planning. Transportation Research Procedia, 2, 264–27

60. Yang C., Kurahashi S., Kurahashi K., Ono I., & Terano T. (2009). Agent-Based simulation on women's role in a family line on civil service examination in Chinese history. JASSS - The Journal of Artificial Societies and Social Simulation, 12(2), 5.

61. Yasami Y., & Mozaffari S. P. (2010). A novel unsupervised classification approach for network anomaly detection by k-Means clustering and ID3 decision tree learning methods. The Journal of Supercomputing, 53(1), 231–245.

62. 菊地 剛正, 國上 真章, 山田 隆志, 高橋 大志, & 寺野 隆雄. (2016). エージェントシミュレーションを用いた金融規制が金融機関の連動的な破綻に与える影響の分析. 人工知能学会論文誌, 31(6), 1–11.

63. 菊地 剛正, 國上 真章, 山田 隆志, 高橋 大志, & 寺野 隆雄. (2016). エージェントシミュレーションを用いた中央銀行の資金供給が金融機関の連動的な破綻に与える影響の分析. 経営情報学会誌, 25(3), 1–17.

64. 坂平 文博, & 寺野 隆雄. (2014). 弥生農耕文化の「主体」は誰だったか？ ―人類学・考古学へのエージェントベースシミュレーションの適用―. コンピュータ ソフトウェア, 31(3), 97–108.

65. 寺野 隆雄. (2003). エージェントベースモデリング : KISS 原理を超えて(<特集>複雑系と集合知). 人工知能学会誌, 18(6), 710–715.

66. 寺野 隆雄. (2010). なぜ社会システム分析にエージェント・ベース・モデリングが必要か. 横幹, 4(2), 56–62.

67. 寺野 隆雄. (2012). 社会シミュレーション技術をいかに納得させるか. 学術の動向, 17(2), 45–47.

68. 出口 弘. (2013). 社会シミュレーション&サービスシステムが目指す世界. 計測と制御 = Journal of the Society of Instrument and Control Engineers, 52(7), 563–567.

69. 松島 裕康, 内種 岳詞, 辻 順平, 山下 倫央, 伊藤 伸泰, & 野田 五十樹. (2016). 実験計画法による実験数削減と有意なパラメータ探索の避難シミュレーション分析への適用. 人工知能学会論文誌, 31(6), 1–9.

70. 倉橋 節也. (2013). 社会システムの研究動向 4-評価・分析手法(2)-モデル推定と逆シミュレーション手法. 計測と制御 = Journal of the Society of Instrument and Control Engineers, 52(7), 588–594.

71. 宝月 誠. (2004). 逸脱とコントロールの社会学—社会病理学を超えて. 有斐閣.

72. 和泉 潔, 斎藤 正也, & 山田 健太. (2017). マルチエージェントのためのデータ解析（マルチエージェントシリーズ). コロナ社.

73. 和泉 潔, 池田 竜一, 山本 仁志, 諏訪 博彦, 岡田 勇, 磯崎 直樹, & 服部 進. (2013). 可能世界ブラウザとしてのエージェントシミュレーション : ターゲットマーケティングへの応用. 電子情報通信学会論文誌, 96(12), 2877–2887.

業績リスト

査読付き論文

・田中祐史, 國上真章, 寺野隆雄, **エージェントシミュレーションにおけるログクラスター の系統的分析からわかること,** JASAG 日本ゲーミングシミュレーション学会 Vol. 27, No. 1, June, 2017

査読つき国際会議

・Yuji Tanaka, Susumu Aida, Takao Terano, **Mining massive logs generated from runs of agent-based simulation.** The Proceedings of Internatinal Workshop: Artificial Intelligence of and for Business 2016 (AI-Biz2016) November 14, 2016, Kanagawa, Japan #8, 10pp.

Yuji Tanaka, Takamasa Kikuchi, Masaaki Kunigami, Takashi Yamada, Hiroshi Takahashi, Takao Terano, **Classification of outputs using log clusters in agent simulation**. The Proceedings of Internatinal Workshop: Artificial Intelligence of and for Business 2017 (AI-Biz2017)

査読なし国内会議

・田中祐史, 菊地剛正, 國上真章, 山田隆志, 高橋大志, 寺野隆雄, **エージェントシミュレーションにおけるログクラスターを利用したシミュレーション結果の分類** 人工知能学会 第 7 回 経営課題に AI を! ビジネス・インフォマティクス研究会, 2017.

謝辞

本論文の作成にあたり，多くの方々からご指導とご支援をいただきましたこと，この場をお借りして厚くお礼申し上げます.

東京工業大学名誉教授である寺野隆雄先生には，本年の３月にご退官されるまで指導教官として，その後も副査として，修士課程と合わせて 5 年間以上ご指導いただいたこと深く感謝いたします. 本研究の問題意識も修士論文に端を発しており, 私自身が気づいていない点を含め, 寺野先生から本研究を進めるうえで，多大なる影響を受けました.

出口弘教授には，4 月から指導教員を引き受けてくださり，研究室に所属させていただき研究の環境を与えていただいただけでなく，数多くのご助言を頂きましたこと深く感謝いたします．私の研究内容について何度も議論をさせていただき，研究の価値づけや，その見せ方について，研究を良くまとめるためのご助言を数多くいただきました.

連携教授である吉川厚先生には，研究をまとめ上げるにあたり，様々な視点からのご助言を何度もいただきましたこと感謝いたします．核心をつくご指摘を頂く度，自身の研究を振り返ることができました.

特別研究員である國上真章さんには，毎週土曜日のゼミにて研究内容についてだけでなく，研究のまとめ方などの作法についても, こと細かくご指導いただきましたこと感謝いたします. 土曜ゼミに参加されている寺野研究室ＯＢの方々，博士課程の方には毎週有益なアドバイスをいただき感謝いたします.

今後は，本研究の過程で得られた知見，学びをさらに深めていくとともに，学術界内外への貢献を目指す所存です.

最後に，博士課程の生活をサポートし続けてくれた妻に感謝いたします.

2018 年 8 月末日

付録

提案手法の python による実装コード(python3)

関数一覧

提案手法全体の関数：

propposed_method()

propposed_method 内で利用している関数：

dtw_distance()
levensitain_dintance()
hierarical_clustering()
calc_ch()
decision_tree()

各関数の説明

propposed_method()

概要：

提案手法，各種の関数を呼び出す.

引数：

モデルが扱う現象についてのプロセスログ（時系列）
モデルが扱う現象についてのプロセスログ（文字列）
エージェントの行動ログ

返り値：

ログごとのクラスタリング結果
得られた決定木

dtw_distance()

概要：

ログ間の距離を DTW 距離により計算し，距離行列を返す.

引数：

モデルが扱う現象についてのプロセスログ（時系列）

返り値：

ログ間の距離行列

levenshtein_dintance()

概要：

ログ間の距離を Levenshtein 距離により計算し，距離行列を返す.

引数：

モデルが扱う現象についてのプロセスログ（文字列）

返り値：

ログ間の距離行列

hierarical_clustering()

概要：

階層的クラスタリングを実行する.

入力：

ログ間の距離行列

クラスター数

返り値：

ログごとのクラスター番号

calc_ch_score ()

概要：

CH 基準により，クラスター数を決定する.

引数：

階層的クラスタリングの結果

調査範囲

返り値：

CH 基準で決まる最適なクラスター数

decision_tree()

概要：

決定木分析を実行する.

引数：

ログごとのクラスター番号

エージェントの行動ログ

返り値：

得られた決定木

各関数の実装コード

```
def proposed_method(emergence_log_ts,emergence_log_st, agents_log,ts_is_true=True):
        """
        提案手法，各種の関数を呼び出す.

        Parameters
        ----------
        emergence_log_ts : double[][]
                モデルが扱う現象についてのプロセスログ（時系列）
        emergence_log_st : String[]
                モデルが扱う現象についてのプロセスログ（文字列）
        agents_log : String[][]
                エージェントの行動ログ
        ts_is_true : bool
                emergence_log が時系列の時は True,文字列の時は False（default=True）

        Returns
        -------
        rt_cluster_per_log : int[]
                ログごとのクラスタリング結果
        tree_graph : String
                得られた決定木
        """

        if ts_is_true:
                dist_mat = dtw_dist_mat(emergence_log_ts)
        else:
                dist_mat = dtw_dist_mat(emergence_log_st)

        rt_cluster_per_log = hierarchical_clustering(dist_mat,2)

        tree_graph = decision_tree(rt_cluster_per_log,agents_log)

        return rt_cluster_per_log,tree_graph
```

```
def dtw_dist_mat(emergence_log):
        """
        ログ間の距離を DTW 距離により計算し，距離行列を返す.

        Parameters
        ----------
        emergence_log : double[][]
                モデルが扱う現象についてのプロセスログ（時系列）

        Returns
        -------
        dist_mat : double[]
                ログ間の距離行列
        """

        n=len(emergence_log)
        dist_mat = []

        for i in range(n):
                for j in range(i+1,n):
                        dist_mat.append(dtw_distance(outputs1[i],outputs1[j]))

        return dist_mat
```

```
def levenshtein_dist_mat(emergence_log):
        """
          ログ間の距離を Levenshtein 距離により計算し，距離行列を返す.

          Parameters
        ----------
        emergence_log : String[]
                モデルが扱う現象についてのプロセスログ（文字列）

        Returns
        -------
        dist_mat : double[]
                ログ間の距離行列
        """

        import Levenshtein
        n=len(emergence_log)
        dist_mat = []

        for i in range(n):
                for j in range(i+1,n):
                        dist_mat.append(Levenshtein.distance(data[i][0], data[j][0]))
          return dist_mat
```

```
def hierarchical_clustering(dist_mat, cluster_n = 0):
        """
        階層的クラスタリングを実行する.

        Parameters
        ----------
        dist_mat : double[]
                ログ間の距離行列
        cluster_n : int
                分割するクラスター数(default=0)
                0 の時は calc_ch_score()によって自動決定される.

        Returns
        -------
        cluster_per_log : int[]
                ログごとのクラスタリング結果
        """

        from matplotlib.pyplot import show
        from scipy.cluster.hierarchy import linkage, dendrogram, ward,cut_tree



        plt.clf
        result = linkage(dist_mat, method='ward')
        dendrogram(result)
        show()

        if cluster_n==0:
                cluster_n = calc_ch_score(dist_mat,result)

        cluster_per_log = flatten(cut_tree(result, cluster_n))
        return cluster_per_log
```

```
def calc_ch_score(dist_mat,clustering_result, n_min = 2, n_max = 10):
        """
        提案手法，各種の関数 CH 基準により，クラスター数を決定する．

        Parameters
        ----------
        dist_mat : double[]
                ログ間の距離行列
        result : linkage
                階層的クラスタリングの結果
        n_min : int
                クラスター数調査範囲の最小値(default=2)
        n_max
                クラスター数調査範囲の最大値(default=2)

        Returns
        -------
        max_cluster_n : int
                得らえれたクラスター数
        """
        from sklearn.metrics import silhouette_score
        from scipy.cluster.hierarchy import cut_tree
        import scipy.spatial.distance as distance

        max_silhouette = 0
        max_cluster_n = 0
        for cluster_n in range(n_min,n_max):
          silhouette_avg = silhouette_score(distance.squareform(dist_mat),
flatten(cut_tree(clustering_result, cluster_n)),metric="precomputed")
          print("For n_clusters =", cluster_n,"The average silhouette_score is :", silhouette_avg)
          if max_silhouette<silhouette_avg:
                max_cluster_n = cluster_n

        return max_cluster_n
```

```
def decision_tree(cluster_per_log,agents_log):
        """
        決定木分析を実行する.

        Parameters
        ----------
        cluster_per_log : int[]
                ログごとのクラスター番号
        agents_log : String[][]
                エージェントの行動ログ
        Returns
        -------
        tree_graph : String
                得られた決定木
        """
        from sklearn import tree
        clf = tree.DecisionTreeClassifier(max_depth=3)
        clf = clf.fit(agents_log, cluster_per_log)

        # 作成したモデルを用いて予測を実行
        predicted = clf.predict(x)
        # 予測結果
        predicted
        # 識別率を確認
        test = sum(predicted == cluster_per_log) / len(cluster_per_log)
        print("Accuracy:",test)

        # 作成した決定木を可視化（pydotplus パッケージを利用）
        import pydotplus
        from sklearn.externals.six import StringIO
        dot_data = StringIO()
        tree.export_graphviz(clf, out_file=dot_data)
        tree_graph = dot_data.getvalue()
        return tree_graph
```

```
def dtw_distance(ts_a, ts_b, d=lambda x, y: abs(x-y), window=0):
    """
    2 つの時系列の距離を計算する DTW の実装.

    Parameters
    ----------
    ts_a : double[]
        比較する一方の時系列 a
    ts_b : double[]
        比較する一方の時系列 b
    d : lambda
        2 変数間の距離の定義(default=lambda x, y: abs(x-y))
    window : int:
        ウィンドウサイズ(default = 0)

    Returns
    -------
    rt_dist : double
        2 つの時系列間の距離.
    """

    import numpy as np
    if window <= 0:
        window = max(len(ts_a), len(ts_b))

    ts_a_len = len(ts_a)
    ts_b_len = len(ts_b)

    cost = np.empty((ts_a_len, ts_b_len))
    dist = np.empty((ts_a_len, ts_b_len))

    cost[0][0] = dist[0][0] = d(ts_a[0], ts_b[0])

    for i in range(1, ts_a_len):
```

```
            cost[i][0] = d(ts_a[i], ts_b[0])
            dist[i][0] = dist[i-1, 0] + cost[i, 0]

        for j in range(1, ts_b_len):
            cost[0][j] = d(ts_a[0], ts_b[j])
            dist[0][j] = dist[0, j-1] + cost[0, j]

        for i in range(1, ts_a_len):
            windowstart = max(1, i-window)
            windowend = min(ts_b_len, i+window)
            for j in range(windowstart, windowend):
                cost[i][j] = d(ts_a[i], ts_b[j])
                dist[i][j] = min(dist[i-1][j], dist[i][j-1], dist[i-1][j-1]) + cost[i][j]
        rt_dist = dist[ts_a_len-1][ts_b_len-1]

        return rt_dist


def flatten(nested_list):
        """2 重のリストをフラットにする関数"""
        return [e for inner_list in nested_list for e in inner_list]
```