

『Trust-Distrust モデル』によるイノベーションの普及要因分析

2024 年 1 月

鳥取大学大学院 工学研究科
機械宇宙工学専攻 博士後期課程

藤居 誠

目次

概要	5
第1章 研究の背景と目的	8
1.1 背景	8
1.2 目的	9
1.3 分析・シミュレーションツール	10
1.4 論文構成	11
第2章 オピニオンダイナミクスと関連分野の研究	13
2.1 社会物理学	13
2.2 オピニオンダイナミクス	13
2.3 ネットワーク, グラフ理論	18
2.4 イノベーション普及の研究	22
2.5 マーケティング・ミックス・モデルと広告効果	26
第3章 エージェントと普及/伝播の対象に関する議論	31
3.1 2カテゴリーエージェントによるシミュレーション	32
3.1.1 最初から興味をもって市場に存在する消費者の操作	32
3.1.2 マスメディア効果の操作	36
3.1.3 ランダムネットワークの操作によるシミュレーション	37
3.2 5カテゴリーエージェントの拡張	41
3.2.1 5エージェントモデルによるシミュレーション	42
3.2.2 態度の初期分布を操作したシミュレーション	43
3.2.3 信頼係数 D_{ij} の操作	45
3.2.4 マスメディア効果の操作	46
3.3 まとめ	48
第4章 普及/伝播のネットワークに関する議論	51
4.1 シミュレーション1: ベース	51
4.1.1 ランダムネットワークを使用したシミュレーション	52

4.1.2	スケールフリーネットワークを利用したシミュレーション.....	52
4.1.3	スモールワールドネットワークを利用したシミュレーション.....	54
4.1.4	シミュレーション1の比較.....	55
4.2	シミュレーション2：シミュレーション1に広告変数を加味したシミュレーション.....	56
4.2.1	ランダムネットワークを使用したシミュレーション.....	56
4.2.2	スケールフリーネットワークを使用したシミュレーション.....	57
4.2.3	スモールワールドネットワークを使用したシミュレーション.....	58
4.2.4	シミュレーション2の比較.....	59
4.3	シミュレーション3：5アダプターカテゴリーによるシミュレーション.....	60
4.3.1	ランダムネットワークを使用したシミュレーション.....	60
4.3.2	スケールフリーネットワークを利用したシミュレーション.....	61
4.3.3	スモールワールドネットワークを使用したシミュレーション.....	63
4.3.4	シミュレーション3の比較.....	64
4.4	まとめ.....	65
第5章 メディア効果に関する議論.....		67
5.1	リーチパターンについて.....	67
5.1.1	電撃型.....	68
5.1.2	くさび型.....	69
5.1.3	逆くさび・対人影響型.....	69
5.1.4	短期流行型.....	70
5.2	シミュレーションの実行.....	71
5.2.1	1カテゴリーエージェントを採用したシミュレーション.....	71
5.2.1.1	マスメディア変数ゼロのシミュレーション.....	71
5.2.1.2	電撃型によるシミュレーション.....	72
5.2.1.3	くさび型によるシミュレーション.....	73
5.2.1.4	逆くさび・対人影響型によるシミュレーション.....	74
5.2.1.5	短期流行型によるシミュレーション.....	75
5.2.1.6	1カテゴリーエージェントのシミュレーション間の比較.....	76
5.2.2	5カテゴリーエージェントを採用したシミュレーション.....	77

5.2.2.1	マスメディア変数ゼロのシミュレーション	78
5.2.2.2	電撃型によるシミュレーション	79
5.2.2.3	くさび型によるシミュレーション	79
5.2.2.4	逆くさび・対人影響型によるシミュレーション	80
5.2.2.5	短期流行型によるシミュレーション	82
5.2.2.6	5カテゴリエージェントによるシミュレーションの比較	83
5.3	まとめ	85
第6章	まとめ	86
6.1	各章で実施したシミュレーション	86
6.2	本研究の貢献	88
6.3	今後の課題	89
謝辞	90	
参考文献	91	
付録	95	

概要

本稿の目的は、イノベーションの普及要因について、人々の意見分布の変動に関する理論を取り扱うオピニオンダイナミクスを用いて検討することにある。イノベーションの普及、つまり消費者が対象となる新製品や新サービスといったイノベーションを採用するためには、まず消費者が当該イノベーションに対して正の態度を形成する必要がある。

オピニオンダイナミクスの数理モデルは多く存在するが、本稿ではエージェント間に信頼と不信の両方が含まれ、かつメディアの影響を扱うことのできる Trust-Distrust モデルを用いている。そこで扱われるエージェントの信頼係数と革新性、ネットワーク構造、そしてメディア変数の多寡やリーチパターンの操作によって生じる変化、態度形成を分析対象としている。

エージェントについては、ロジャースに代表される新製品や新サービスといったイノベーションの普及として多くの研究者に認識されている採用者カテゴリーを参考にしている。採用者カテゴリーとは、個々人の革新性に基づいて社会システムの成員を区別したものである。また革新性とは社会システムに属する他の成員と比較して、自身が新しいアイデアや製品を相対的に早期に採用する度合いである。ロジャースはこの革新性に基づき、社会システム内の個人をイノベーションが社会に普及・拡散する過程に即して5つのカテゴリーに分類している。各採用カテゴリーはイノベーションの採用順に「イノベーター」「アーリーアダプター」「アーリーマジョリティ」「レイトマジョリティ」そして「ラガード」である。

Trust-Distrust モデルのシミュレーションで利用されるネットワーク構造は、ランダムネットワーク、スケールフリーネットワーク、そしてスモールワールドネットワークである。複数のネットワーク構造を適用させることで、ネットワークの違いやそこで適用されるその他の変数の違いによるイノベーションの普及、消費者の態度形成について検討している。

またメディア変数は、シミュレーション内では外圧として機能し、エージェントをメディア変数の導く方向に向かわせる。つまり広告といったメディアに接触する

とそれを認知し態度変容を起こすと考える。そこでメディア変数のリーチパターンを電撃型、くさび型、逆くさび型、そして短期流行型と変化させ、広告接触後の態度変容について観察する。

本稿では、社会物理学のみならずマーケティング、グラフ理論などの知見からモデル構築とシミュレーションを実行している。これにより、シミュレーションと実社会の関係性を深めることができ、多様化・複雑化しているメディアと消費者、そしてエージェント間の関係を俯瞰し、消費者の態度形成やイノベーションの普及過程を把握することを目的としている。これにより、オピニオンダイナミクス理論に関心を有する研究者が今後の研究テーマを検討する際や、実務家が実際のメディア計画を立てる際に有益な情報となることを期待している。

本論文の構成は、以下のようになっている。

第1章：研究の背景と目的

本研究を行う背景と目的と本論文の構成について説明している。

第2章：オピニオンダイナミクスと関連分野の研究

Trust-Distrust モデルのシミュレーションを実行する際に必要となる、イノベーションの普及、グラフ理論、メディア効果について、また関連分野のレビューを行う。

第3章：エージェントと普及/伝播の対象に関する議論

Trust-Distrust モデルのシミュレーションに用いられるエージェントについて、カテゴリ数異なるモデルを構築しシミュレーションを実行する。エージェントのカテゴリ数は、1, 2, 5カテゴリ存在するモデルを構築し、複数のカテゴリが存在するモデルでは、シミュレーションに投下されるタイミングや信頼係数をカテゴリ毎に操作することで、イノベーションの普及に与える影響を確認する。

第4章：普及/伝播のネットワークに関する議論

Trust-Distrust モデルに用いられるネットワークを、ランダムネットワーク、スケールフリーネットワーク、スモールワールドネットワークといったネットワーク構造下でのオピニオンダイナミクスを検証していく。

第5章：メディア効果に関する議論

Trust-Distrust モデルに組み込まれているメディア項について、リーチパターンを考慮したモデルを構築しシミュレーションを実行する。それによって、出稿パターンによってイノベーションの普及に与える影響を把握する。

第6章：結論

各章の内容をまとめ、本論文の寄与について説明する。最後に、本論文の研究成果を踏まえた今後の課題と展望について述べる。

第 1 章 研究の背景と目的

1.1 背景

人々は社会において何らかのつながりをもって生活している。それは家族や友人・知人といった顔の見える物理的なつながりだけではなく、近年の ICT (Information and Communication Technology) の発展により普及した Facebook や X (旧 : Twitter) に代表される様々な SNS (Social Networking Service) によってもつながっている [1] [2] [3].

つながりという点においては、2003 年に報告された SARS (重症急性呼吸器症候群) や 2023 年 5 月 8 日より「新型インフルエンザ等感染症 (いわゆる 2 類相当)」から「5 類感染症」になったことでメディアの露出機会は減ったが、現時点においても世界的に感染が蔓延している新型コロナウイルス感染症 (COVID-19) の感染対策としても、人と人の接触を伴う物理的なつながりに注目が集まっている。感染症は物理的な接触 (飛沫感染, 空気感染, 接触感染など) において引き起こされるが、新製品やサービスは物理的な接触のみならず SNS などによる e クチコミによっても伝播し、その影響は計り知れない [4] [5] [6].

これらのつながりはネットワークと呼ばれネットワークとグラフ理論として研究されている。感染症の広がりを表すネットワーク上の伝播モデルとしては、SIR(Susceptible Infected Recovered)モデルがよく利用される [7]. また社会の人々が意見を交わしながら合意形成を目指す動きを記述するモデルとしては、オピニオンダイナミクスが知られている。オピニオンダイナミクスは、多くの人々が周囲の人との相互作用によって合意形成していく様子を離散写像でモデリングする社会物理学の研究テーマの一つである [7] [8] [9] [10] [11] [12] [13].

マーケティングにおいては、従来型のマスメディアに加えてデジタルメディアも活用されている。デジタルメディアでは、従来型のメディアのように情報を消費者に

ダイレクトに伝えることを目的としているものから、ソーシャルメディアを通じた消費者間のインタラクションによる情報の伝播を期待して利用されることも多い [7]. 新商品や新サービスといったイノベーションが市場に浸透していく過程の本質は情報交換であり、情報交換を通じて人々は新しいアイデア知ることとなる. 新製品や新サービスが上市されたという情報を消費者に最も早く効率的に伝達するのに適した方法はマスメディアである. そして人を説得して新しいアイデアを受入れさせるには、対面での情報交換が有効だとされている [14]. マスメディアだけではなく、人々のインタラクションも加味した情報拡散を研究していく意義は大きいといえる.

1.2 目的

本稿の目的は、イノベーションを消費者に採用させるため、消費者の態度形成について、人々の意見分布の変動に関する理論を取り扱うオピニオンダイナミクスを用いて検討することにある.

そのため、オピニオンダイナミクスの一つであり、信頼と不信の両方が含まれ、かつメディアの影響を扱うことのできる **Trust-Distrust** モデルに着目し、そこで扱われるエージェント、ネットワーク、そしてメディア変数の操作によって生じるシミュレーション結果の差異を分析対象とすることとした. 情報をもたらす他者への信頼や不信は、消費者自身が態度を形成する際大きく影響をあたえるので、信頼と不信の関係を扱える **Trust-Distrust** モデルを利用することのメリットは大きいといえる. またシミュレーションを実行するメリットとしては、実務において広告キャンペーンを展開する前に、実際に市場調査や実験を行うことが困難状況でも、設定を色々に変更しコンピュータ上で再現し検証することが可能な点が挙げられる.

エージェント¹については、ロジャースに代表されるイノベーションの普及として多くの研究者に認識されている採用者カテゴリーを参考にしている. 採用者カテゴリー

¹ 一定のルールで自律的に行動する主体. 本稿では消費者にあたる.

一とは、個々人の革新性に基づいて社会システムの成員を区別したものであり、革新性とは社会システムに属する他の成員と比較して、自身が新しいアイデアや製品を相対的に早期に採用する度合いとしている [14]. ロジャースはこの革新性に基づき、社会システム内の個人を、イノベーションが社会に普及・拡散する過程を5カテゴリーに分類している. 各採用者カテゴリーはイノベーションの採用順に「イノベーター」「アーリーアダプター」「アーリーマジョリティ」「レイトジョリティ」そして「ラガード」である [14].

Trust-Distrust モデルのシミュレーションで利用されるエージェント間のネットワークは、ランダムネットワーク、スケールフリーネットワーク、スモールワールドネットワークを適用させる. ネットワークの種類を変えることでシミュレーション結果に違いが生じるのか検証する.

またメディア変数は、シミュレーション内では外圧として機能し、エージェントをメディア変数の導く方向に向かわせる. そこでメディア変数の投下パターンを、広告の実務で利用されているメディアのリーチパターンにあわせてシミュレーションすることで、イノベーションの普及について観察する.

本論文では、社会物理学のみならずマーケティング、グラフ理論などの知見からモデルの構築とシミュレーションを実行している. これにより、シミュレーションと実社会の関係性を深めることが出来、多様化・複雑化しているメディアとエージェント（消費者）、そしてエージェント間の関係を俯瞰し、イノベーションの普及過程を把握することを目的としている. これにより、オピニオンダイナミクス理論に関心を有する研究者が今後の研究テーマを検討する際や、実務家が実際のメディア計画を立てる際に有益な情報となることを期待している.

1.3 分析・シミュレーションツール

Trust-Distrust モデルについては、Web アプリケーションである Jupyter Notebook か

らプログラミング言語 Python 3.7.3 を用いて、分析とシミュレーションを実行した。このプログラミング言語は、Guido van Rossum によって開発され、現在 Python Software Foundation によってプロモートされている [15][16]。また VAR (ベクトル自己回帰) モデルと時変係数モデルの構築では、オープンソース・フリーソフトウェアの統計解析プログラミング言語である R (version 4.0.5) を利用した。この言語はニュージーランドのオークランド大学の Ross Ihaka と Robert Clifford Gentleman により開発され、現在 The R Foundation によってサポートされている [17]。

1.4 論文構成

本論文の構成は、以下のようになっている。

第1章：研究の背景と目的

研究の背景と目的について論じる他、本論文で使用した分析・シミュレーションについて、そして本論文の構成について記述している。

第2章：オピニオンダイナミクスと関連分野の研究

本研究で扱う Trust-Distrust モデルの他、ネットワーク、そしてエージェントのカテゴリ化で扱うイノベーションの普及研究についてのレビューを行う。

第3章：エージェントと普及/伝播の対象に関する議論

Trust-Distrust モデルのシミュレーションに用いられるエージェントについて、カテゴリ数の異なるモデルを構築しシミュレーションを実行する。カテゴリはエージェントの革新性によって区別され、エージェントのカテゴリ数は、1, 2, 5 カテゴリ存在するモデルを構築する。複数のカテゴリが存在するモデルでは、シミュレーションに投下されるタイミングや信頼係数をカテゴリ毎に操作することで、イノベーションの普及に与える影響を確認する。

第4章：普及/伝播のネットワークに関する議論

Trust-Distrust モデルに用いられるネットワークは、ランダムネットワーク、スケー

ルフリーネットワーク, スモールワールドに変更したモデルを構築しシミュレーションを実行する.

第5章：メディア効果に関する議論

Trust-Distrust モデルに組み込まれているメディア項について, リーチパターンを考慮したモデルを構築しシミュレーションを実行する. それによって, 出稿パターンの違いが, イノベーションの普及に与える影響を把握する.

第6章：まとめ

各章の内容をまとめ, 本論文の寄与について説明する. 最後に, 本論文の研究成果を踏まえた今後の課題と展望について述べる.

第2章 オピニオンダイナミクスと関連分野の研究

2.1 社会物理学

物理学は、落下運動、振動、電流、熱などといった現象を統一的に説明できる理論を背景にもっている。社会物理学は社会現象の特徴的な動きを統一的に記述できる何らかの法則があると考え、物理学の手法を用いて社会現象の特徴的な動きを探っている学問とすることができる。その歴史は浅く、基本法則が見つかったわけではないが、社会現象を統一的に説明できる理論が背景にあることが望ましいと考えられている。社会物理学の研究対象としては、街中での人や車の流れ、お金や情報の流れなどがある。特にお金の流れに焦点を絞った研究は、経済物理学と呼ばれている。

社会物理学の研究手法のポイントは、統計学のように社会データを統計的に処理するだけではなく、社会の背後に基本法則があると考え社会の動きを統計的に捉えるだけではなく、社会現象には因果関係があると考え、因果関係には原因と結果という時間軸が入ってくるので、時間依存の数理モデルで現象が記述できると考える。そして、いくつかの独立と思われる社会現象を統一的に記述できる数理モデルが提案されれば、社会を記述する統一法則に少しは近づいたと考える。統計的な精度よりも動きを記述することを優先する場合もある。社会現象の特徴的な動きを記述する場合、物理学者が慣れ親しんだ微分方程式を用いて現象を記述する数理モデルを提案することが多い [7]。

2.2 オピニオンダイナミクス

社会物理学の研究テーマの一つにオピニオンダイナミクスがある。多数の人が周囲の人々と相互作用しながら意見を変え、社会的合意形成を目指していくプロセスを離散写像でモデル化し解析する理論として、様々な面から研究がなされている [7] [8] [9] [10] [11] [12] [13]。

オピニオンダイナミクスのモデルとしては、人々の意見を2値（0 か 1 または-1 か 1）の離散的な値とする数理モデルと、人々の意見を数値化し連続的に分布すると考えるモデルの2つが知られている [18].

フランスやアメリカの大統領選挙、ブレグジットでみられた国民投票では、離散的な2項理論を適用することが出来る。2値理論の代表的な数理モデルとしては、Granovetterによって提案されたVoterモデル [19][20][21]の他、Galamの磁気物理学の理論 [22]や局所多数派理論 [23][24]があげられる。

Voterモデルには多くの発展形が提案されているが、基本は選挙での投票行為に代表される2択（採用する/しない）の記述を目的にしている。標準的なVoterモデルでは、意見は $S = \pm 1$ とされる。個人 i の意見を S_i 、個人 j の意見を S_j とする。全体を N 人とし、全体での意見の集合を $S = \{S_i\}$ とする。また S と S_k の値だけ異なっている意見の集合を S^k とする。すると、時刻 t での全体の意見分布が S となる確率を $P(S, t)$ とした場合、以下の方程式で求められる。

$$\frac{d}{dt}P(S, t) = \sum_k [W(S^k)P(S^k, t) - W(S)P(S, t)] \quad (1)$$

ここで、 $W(S^k)$ は S^k のうち S_k を反転させる確率を意味している。このような意見の値を2項とする数理モデルは、2項を磁石のS極とN極という二値に対応させて考えることができ、磁性物理学のIsingモデルと本質的に同じとなる。

一方、意見を0から1の連続値で表現するオピニオンダイナミクス理論の数理モデルは、Bounded Confidence Model [25][26][27]と呼ばれ、人々が意見交換によって合意の妥協点を見つけることが前提となっている。主だったモデルとしては、The Deffuant - Weisbuch Model [28][29]と The Heselmann - Krause Model [30]が知られている。Bounded Confidence Model [25][26][27]の基本的な考え方は、個人 i が周囲の人々の意見に影響され、自身の意見も変わるというものである。

The Heselmann - Krause Model は以下で定義される。

$$x_i(t+1) = \sum_{j=1}^N D_{ij}x_j(t) \quad (2)$$

ここでは N 人を想定し、 $1 \leq i \leq N$ として時点 t での個人 i の意見を $x_i(t)$ と書く。ただし意見 x_i の範囲は $0 \leq x_i(t) \leq 1$ である。係数 D_{ij} は N 人中の全ての組合せ i, j について正の実数で様々な値をとる他者から影響を受ける係数で、 $D_{ij} = 0$ ならば個人 i は個人 j の意見を無視することを意味する。また意見の閾値が $0 \leq x_i(t) \leq 1$ であることから、このモデルでは係数 D_{ij} が負の値を取ることを想定していない。意見は0から1の間の連続値を取り（1が同意、0が無関心）、反対意見の仮定はない。つまり暗黙に合意形成を前提としていることが見て取れる。

しかしながら我々は、現実の社会において人々の合意形成が時として大変難しいケースもありうることを知っている。そこで石井-川畑 [31]、石井 [32]は合意形成するのが難しい問題に対処するために、The Heselmann-Krause Model [30]を拡張し、オピニオンダイナミクス理論に人々の中の信頼の欠如を含めたモデルを提唱している。石井らは係数 D_{ij} の意味を信頼係数と修正し、個人 i と個人 j の間に信頼関係があるなら $D_{ij} > 0$ 、不信の関係なら $D_{ij} < 0$ と仮定している。そして人々は自分たちの意見から遠く離れた意見に対して同意または反発することなく無視するものと見なし、彼/彼女自身に非常に近い意見の影響は特に受けないと仮定している。この2つの過程を含め、石井らはThe Heselmann-Krause Model で用いられた $D_{ij}x_j(t)$ 項の代わりに、以下の関数を使用している。

$$D_{ij}\varphi(I_i, I_j)(I_j(t) - I_i(t)) \quad (3)$$

ここで

$$\varphi(I_i, I_j) = \frac{1}{1 + \exp(\beta(|I_i - I_j| - b))} \quad (4)$$

はカットオフが働く関数として機能している。また時点 t での個人 i の意見は $I_i(t)$ 、個人 j の意見は $I_j(t)$ である。

さらに石井らの提唱しているモデル [31] [32]では、メディアの影響とトピック

の忘却が加味されている。人間同士の接触（情報交換）の他にも、メディアからもたらされる情報は、世論形成や人々の意見を左右するため、オピニオンダイナミクス理論としても重要な項となる。また時間の推移を考慮した際、問題とされている事項そのものが古くなり人々の関心が薄れていく効果を考慮することもオピニオンダイナミクス理論としては重要なポイントと考えられる。

メディア効果に関しては、 $A(t)$ が時間 t における外部からの圧力、そして各エージェントの反応差が係数 c_i で表されている。係数 c_i はひとり一人に異なる値を持つことができ、 c_i は正または負になることができる。係数 c_i が正の場合、エージェント i はメディアの方向に向かって意見を移動する。逆に係数 c_i が負の場合、メディアの方向に逆らって人の意見が変わる。また指数減衰関数を導入することで、問題の忘却に対応している。この2つの効果を含めると、エージェントの意見の変化は次のように表すことができる。

$$\Delta I_i(t) = -\alpha I_i(t)\Delta t + c_i A(t)\Delta t + \sum_{j=1}^N D_{ij}\varphi(I_i(t), I_j(t))(I_j(t) - I_i(t))\Delta t \quad (5)$$

ここで仮にエージェント i と j が対象となるトピックに対して正の意見をもっていたとしても、互いに不信感をもっている場合 ($D_{ij} < 0$, $D_{ji} < 0$)、信頼係数 D_{ij} が-1から0の間でランダムに決定されるので、エージェント i か j の意見が負になる可能性は否定できない。これは「坊主憎けりゃ袈裟まで憎い」ということもあるので、実社会でもままたまあることだといえよう。

しかしながら、このこととは別にエージェントが2名の場合、カットオフ関数 φ が0から1の間の値を取ることと、 D_{ij} が-1から1の間の値を取ることから逆転は発生しないと思われる。つまり $-1 < \varphi D_{ij} < 1$ であれば逆転は発生しない。しかしながらエージェント数が3以上のときは、 φD_{ij} の和を取る形になるので、 ΔI を計算する右辺に制約を加えなければ、正の意見の人同士が出会っているのに ΔI が負になってしまうケースがあり得る。

つまり2エージェントの場合, $\Delta I_i = D_{ij}\varphi(I_i, I_j)(I_j - I_i)$ で, $D_{ij} = 1$ (信頼), $\varphi(I_i, I_j) = 1$ とすると, $\Delta I_i = I_j - I_i$, つまり $I_i(t + \Delta t) = I_i(t) + \Delta I_i = I_j(t)$ となる. よって $I_i(t) > 0$, $I_j(t) > 0$ のときに $I_i(t + \Delta t) < 0$ になることはない. しかし仮に $D_{ij}\varphi(I_i, I_j) = 2$ だとすると, $\Delta I_i = 2(I_j - I_i)$, つまり $I_i(t + \Delta t) = I_i(t) + \Delta I_i = 2I_j(t) - I_i(t)$ で, $I_i = 3$, $I_j = 1$ の場合, $I_i(t + \Delta t) = -1$ となる. i も j も正の意見を持っており, かつ i は j を信頼しているのにも関わらず $I_i(t + \Delta t)$ が負になってしまう状況がある. また3以上のエージェントのときは $D_{ij}\varphi(I_i, I_j) = 1$ の場合でも, $\Delta I_i = \sum_{j=1}^N N(I_j - I_i) = \sum_{j=1}^N (N I_j - N I_i)$, つまり $I_i(t + \Delta t) = I_i(t) + \Delta I_i = \sum_{j=1}^N (N I_j(t) - (N - 1) I_i(t))$ となり, $I_i(t)$ と $I_j(t)$ が全て正でも $I_i(t + \Delta t)$ が負になりえる状況がある. この点についてモデル式の改良が望まれ, 今後の課題の一つと言える.

本論ではエージェントの意見が正の値であるとき, 商品やサービスといった対象となるイノベーションに対して好意的な態度が形成されたと捉え, また負の場合は否定的な態度が形成されたと捉える. 消費者行動における態度とは, 製品やサービスといった対象に対する「好き・嫌い」や「良い・悪い」といった評価を意味している [4] [5]. また青木ら (2012) は, 「あるブランドに対して好意的な態度を持つ消費者は, そのブランドを購入する可能性が高く, また, ブランドに対する好意的な態度は, 広告によって生み出したり変えたりできる」と述べている [5]. こうした態度が実際の行動, イノベーションの採用 (adopt) や拒絶 (reject) につながっていると考えられる.

消費者行動の理論では, 自身の持つ内部情報に外部からもたらされる新しい情報を加味して商品やサービスといったイノベーションに対する態度が形成されると考える. 新しい情報をもたらす外部とは, メディアまたは他者との会話を指している. ケラーとフェイ (2016) によれば, アメリカで製品やブランドについて交わされる会話の90%はオフラインであることをトークトラックという大規模調査で示している [33]. トークトラックが調査対象としている会話とは, 対面, 電話, 電子メール, インスタント・メッセージ, チャット, ブログ, ツイッター, SNS, 及びその他の会話となっ

ている [33]. その点も踏まえ本論では、オンラインとオフラインの両方の会話を、エージェント間の相互作用として捉えることとする.

また消費者がイノベーションに対して好意的な態度を形成したとしても、実際にそれを採用または購入するか否かについては、本人の関与の他、価格や配荷状況など様々な市場要因が存在するので、今回のシミュレーションだけではその判定するのは難しい. そこで本稿では、エージェントがエージェント同士のインタラクション、そして外部からの情報であるメディアからの影響を受け、イノベーションに対する態度変容を Trust-Distrust モデルによるシミュレーションによって観察する.

2.3 ネットワーク, グラフ理論

人やモノなど社会の構成要素は何らかの関係性を持っており、それらを頂点として線をつなぎ幾何学的に表現するとネットワークとなる. 鉄道網は駅を頂点とし、航空網は空港を頂点としたネットワークとすることができる. またインターネットや SNS は、世界規模の情報ネットワークといえる [34] [35] [36]. ネットワークは頂点 (ノード) と、頂点と頂点を結ぶリンク (エッジ) からできている [1] [6] [7] [34] [35] [36]. またネットワークは数学の分野ではグラフ理論として、物事のつながりを表すための理論として研究されてきた [34] [35] [36].

ネットワークには様々な種類が存在している. すべてのノードが他の全てのノードとリンクが張られているネットワークを完全グラフ (図 1 左) という. 2つのノードの間にリンクが張られているかどうかはランダムで、リンクが張られている確率が $0 < p < 1$ となっているグラフのことをランダムグラフ (図 1 右) という. 接続確率 $p = 1$ のランダムグラフは完全グラフとなる.

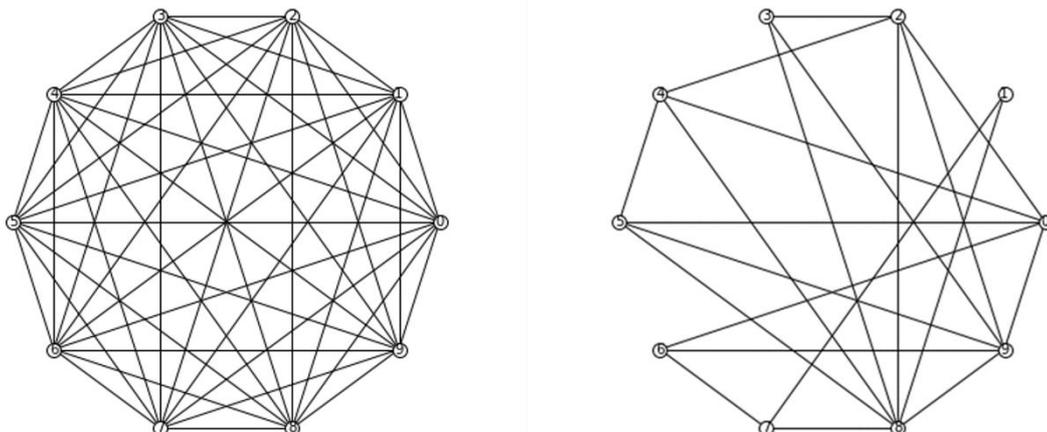


図 1 : 左_N=10 の完全グラフ. 右_N=10, 接続確率 50% のランダムグラフ.

ノードから他のノードに向かうリンクの数のことを次数といい, 各ノードから出ているリンク数の平均を平均次数という. また 3 つのノードが互いにリンクでつながり形成された三角形をクラスタと呼ぶ. ネットワークがどの程度密接に繋がっているかを表す値にクラスタ係数がある. ノード i のクラスタ係数 C_i は各ノードにできる三角形の数とそのノードのリンク数の組合せの比として, 以下で定義される.

$$C_i = \frac{2L_i}{k_i(k_i - 1)} \quad (6)$$

ここで L_i はノード i の k_i 個の隣接ノードの間のリンク数

グラフ理論の定番のグラフとしては, 完全グラフやランダムグラフのほか, スケールフリーネットワークやスモールワールドネットワークなどがある.

スケールフリーネットワークは, 次数分布がべき則で近似できるネットワークである. 電車や飛行機といった交通網に代表され, ネットワークの成長と効率性を考慮した, 現実世界にしばしばみられる他者との結びつきが極めて多い存在をグラフで示したものである. バラバシは交通網の他, インターネットやホームページのリンクの関係はべき則にしたがうことを指摘している [1] [36]. 図 2 は, バラバシ・アルバーモデル [36] というスケールフリー生成モデルを用いて生成したスケールフリーネ

ットワークである. スケールフリーの大きな特徴は次数の大きいハブと呼ばれるノードの存在である. 図2の次数分布を図3に示している.

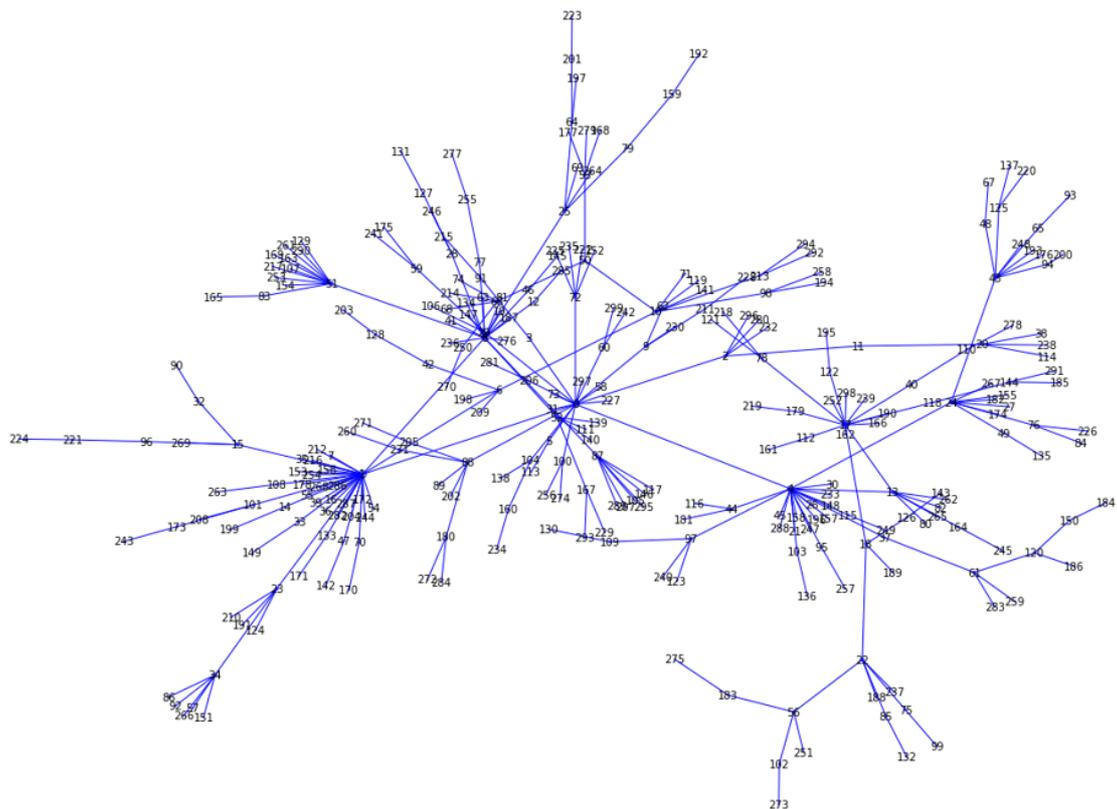


図 2 : スケールフリーネットワーク (バラバシ・アルバートモデル) $N = 300$, 平均次数 = 1

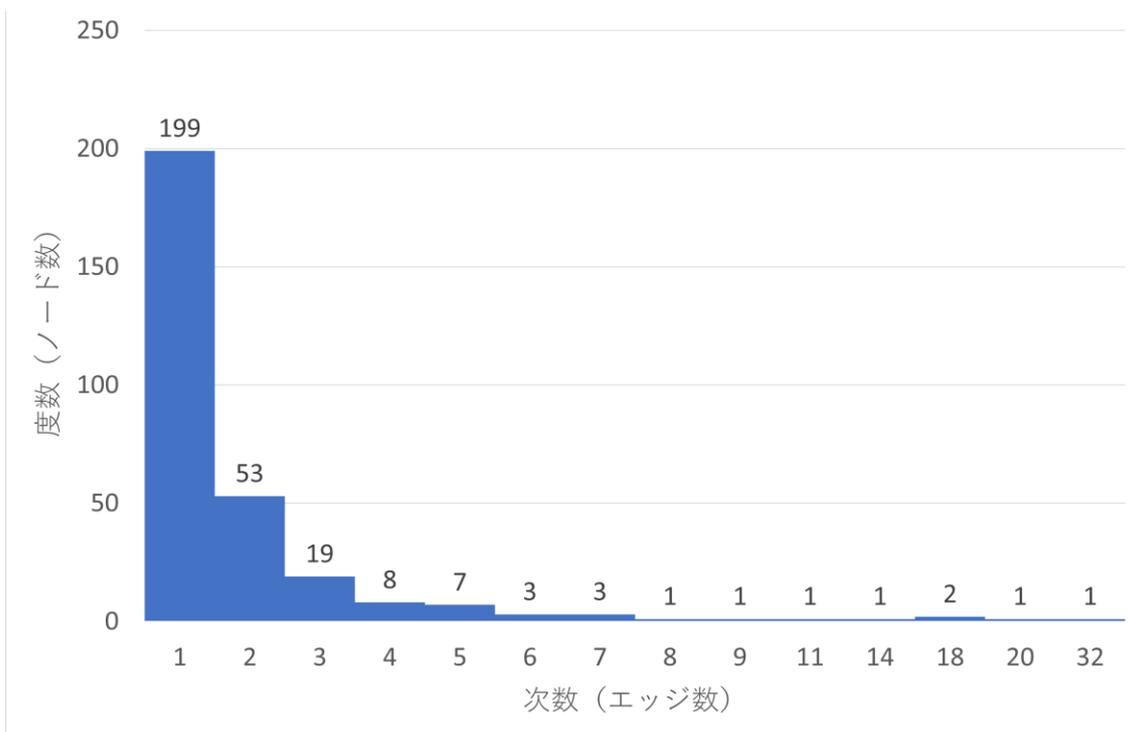


図 3 : スケールフリーネットワーク (図2) の度数と次数のヒストグラム

スモールワールドネットワークは、6次の隔たりというスモールワールド現象で知られており、人間関係や自然界の様々なネットワークを説明するのに使われている [1] [6] [7] [36]。スモールワールドについては、ミルグラムの郵便実験 [1] [6] [36] [37]や Six Degrees of Kevin Bacon [37] [38]など、世間は意外と狭いということを調べたケースが有名である。スモールワールドネットワークの特徴としては、ノード数が多くとも2ノード間の距離を平均した値が短い点、人間関係で例えると、内輪の付き合いが多いという点があげられる。図4は、ワッツ・ストロガノフモデル [36] というスモールワールド生成モデルを用いて生成したスモールワールドネットワークである。また図5に図4の次数分布表を示している。

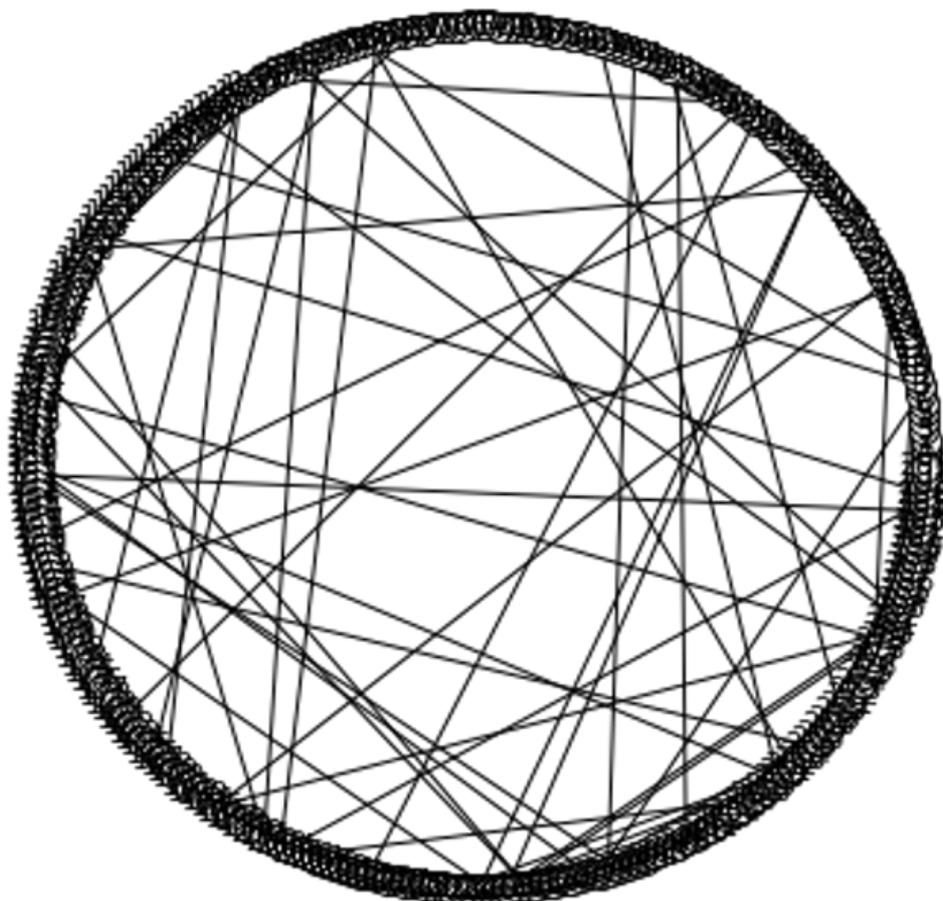


図 4 : スモールワールドネットワーク (ワッツ・ストロガノフモデル)

$$N = 300, \quad k = 4, \quad p = 0.1$$

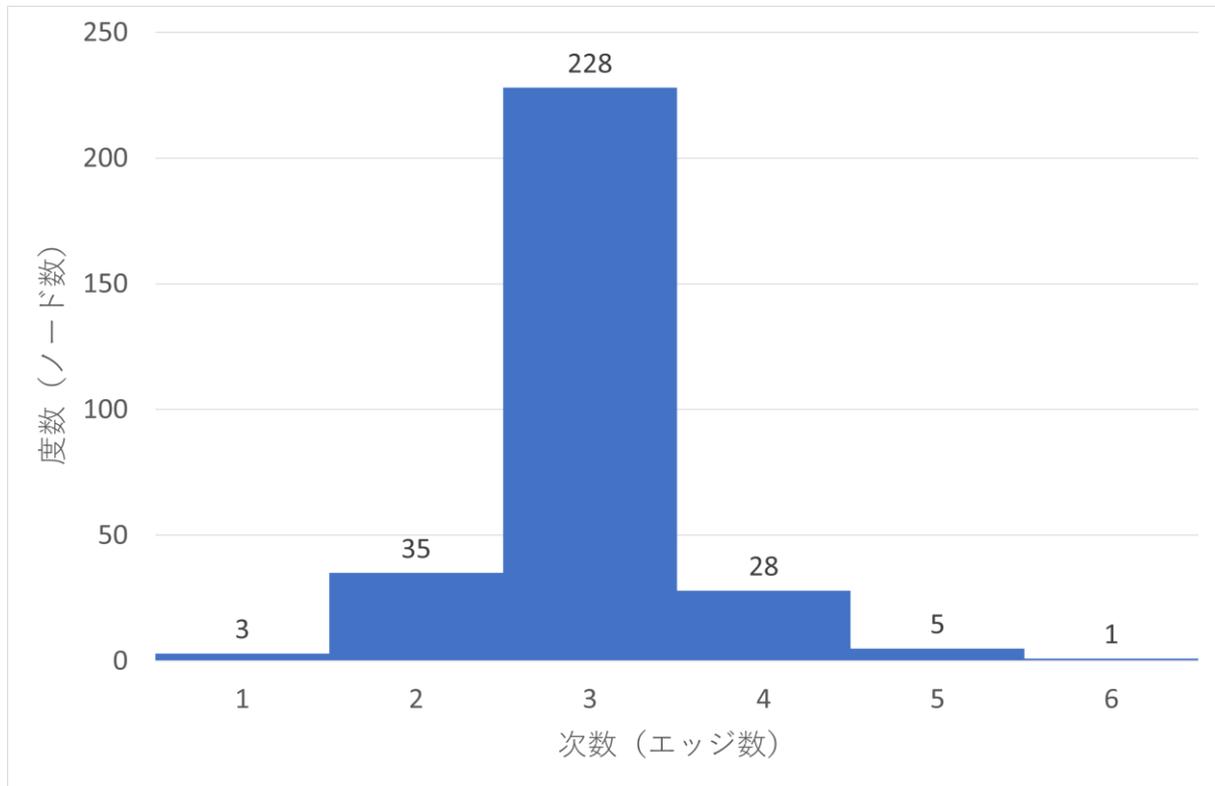


図 5 : スモールワールドネットワーク (図 4) の度数と次数のヒストグラム

オピニオンダイナミクス理論, Trust-Distrust モデルによるシミュレーションでは, メディア変数による外圧とエージェント間の相互作用によって, 集団における態度形成が解析される. そのためシミュレーションに用いられるネットワーク構造は重要なポイントとなる. そこで本論ではランダムネットワーク, スケールフリーネットワーク, スモールワールドネットワークを用いネットワークの違いによる差異についても観察する.

2.4 イノベーション普及の研究

イノベーションとは, 個人あるいは他の採用単位によって新しいと知覚されたアイデア, 習慣, あるいは対象物を指している. また, その普及とはイノベーションが, あるコミュニケーション・チャンネルを通じて, 時間経過の中で, 社会システムの成員間に伝達される過程とロジャースは定義している [14]. マーケティングの文脈で

は、携帯電話カテゴリーにおけるスマートフォン(iPhone)やヨーグルト市場でのギリシャヨーグルト Chobani は、イノベーションの好例であるといえる。

また普及過程の本質は情報交換であり、情報交換を通じて人は新しいアイデア、新商品や新サービスを他の人たちに伝達する。イノベーションが存在しているという情報を潜在的に伝達するのに最も早く効率的な方法はマスメディアであり、人を説得して新しいアイデアを受入れさせるには、対面での情報交換が有効だとされている。対象物の伝播・伝搬という点においては、インフルエンザや新型コロナウイルス感染症(COVID-19)と同様に捉えることも可能かもしれないが、イノベーションを認知したあと、それを自身の判断で採用または拒絶出来るという点においては大きく異なる。採用する/しないの選択において、イノベーションの優劣や需要は普及に大きな影響をあたえるが、本論ではイノベーションの優劣自体は議論せず、オピニオンダイナミクスを活用したイノベーションの普及に焦点を当てる。

イノベーションを採用または拒絶するのは社会システムの成員、マーケティングにおいては市場であり消費者ということが出来る。消費者は日々の生活において発生する様々な問題を解決するため、または自身のニーズを満たすため、既存の製品やサービスはもとより新製品や新サービスといったイノベーションを採用する [39]。また消費者は各人の生活上発生した問題を解決する以外にも、他者の影響を受けて商品やサービスといったイノベーションを採用する場合もある。周囲の影響を受けて需要を形成するバンドワゴン効果 [5] [6] [40]は、消費者間の相互作用の結果といえるので、その典型であるといえる。イノベーションを採用する消費者の分析を行い、マーケティング活動を効果的かつ効率的に推進するためには、消費者の分類を行う必要があり、以下に述べる採用者カテゴリーは消費者を分類する方法の一つといえる。

ロジャースは『イノベーションの普及』 [14]において採用者カテゴリーを提唱しており、消費者が新製品や新サービスといったイノベーションを採用(購入)するタ

イミングは、経済的な要因というより「あたらし物好き」「保守的」といったパーソナリティに依存すると指摘している [14].

その採用者カテゴリーとは、個々人の革新性に基づいて社会システムの成員を区別したものであり、革新性とは社会システムに属する他の成員と比較して、自身が新しいアイデアや製品を相対的に早期に採用する度合いとしている。

採用者カテゴリーは、消費者がイノベーションを採用する順に5つに分類されており、①イノベーター、②アーリーアダプター、③アーリーマジョリティ、④レイトマジョリティ、⑤ラガードとされている(図6)。

革新性に基づいた採用者カテゴリー

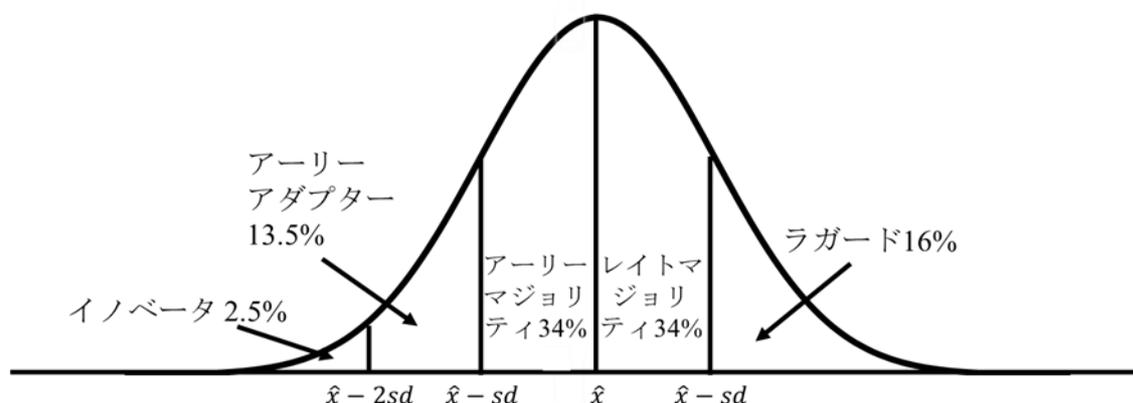


図 6：革新性に基づいた採用者カテゴリー（出典：『イノベーションの普及』(2020)）

イノベーターは冒険的であり、社会システムのゲートキーパーの役割を果たしている。つまり、5つの採用者カテゴリーの中で最も早くイノベーションを採用する層である。アーリーアダプターは、潜在的な採用者にとってイノベーションに関する情報源となるオピニオンリーダー的存在である。アーリーマジョリティはイノベーションの採用に際しては慎重派ではあるが、普及過程におけるつなぎ役であり、対人ネットワークにおける相互連絡役も担う重要なカテゴリーでもある。レイトマジョリティは、イノベーションに対して懐疑的であるといわれており、採用率が50%を超えてからイノベーションを採用する。レイトマジョリティにとって、仲間内の圧力はイノベ-

ション採用を動機づけるのに必要とされている。ラガードは、社会システムのなかで最後にイノベーションを採用するカテゴリーであり、またイノベーションの採用に最も時間の掛かるカテゴリーである。

またマーケティング・サイエンスの領域で、新製品の普及を取り扱ったモデルのうち最も有名なモデルはバス・モデルであるといわれている [6] [41] [42]。ロジャースのモデルが経験則で導き出された近似的な正規分布モデルであるのに対し、バス・モデルではイノベーターとイミテーターそれぞれの採用ペースを表すパラメータ推定を行い新規採用者を割り出している。t期の売上を $n(t)$ とすると、バス・モデルは以下のようなになる。

$$n(t) = [p + qF(t)][1 - F(t)]N \quad (7)$$

ここで

N : 市場規模

$F(t)$: t期までに購入した消費者の割合

p : イノベーター係数

q : イミテーター係数

(7)式にある p , q , N , 3つのパラメータを推定することによって各期の売上予測が可能となる。パラメータの推定には、市場調査やテストマーケティングの結果、担当者の主観、類似製品の値などが使われる [41]。

バス・モデルでのイノベーターは、他者の影響を受けずに新商品を採用するという意味であり（イミテーターは他者から影響を受けて新規採用する消費者）、ロジャースが革新性をもってイノベーターを定義している点とは異なっている。

先行研究における Trust-Distrust モデルでは、エージェントは一つのカテゴリーとして扱われ、シミュレーションのスタート時から全員が一斉にシミュレーションの対象となる。本稿ではイノベーションを採用する採用者カテゴリーの時差と相互作用をも検討するため、ロジャースの採用者分類を採用し、エージェントを分割し時差をも

ってシミュレーションに投下することで、Trust-Distrust モデルにおけるイノベーションに対する好ましい態度と否定的な態度の推移を考察する。またロジャースの5カテゴリーは、採用タイミングの話であるが、本研究では市場に参入する（イノベーションに気付く）タイミングとして援用している。

2.5 マーケティング・ミックス・モデルと広告効果

Trust-Distrust モデルのメディア変数A項は、エージェントの合意形成に影響を与える外圧として取り扱われている。これは他のオピニオンダイナミクスのモデルにはない項で、Trust-Distrust モデルの特徴の一つであるといえる。メディア変数が商品やサービスといったイノベーションの採用に与える影響を分析する手法としては、マーケティングの分野では広告の効果検証として、数理解析的な手法を取り入れたマーケティング・ミックス・モデルが使われている [39] [43] [44] [45] [41]。

マスメディアや広告の効果を取り扱う時系列モデルとしては、VAR (Vector Auto Regression Model) [46] [47] [48]や状態空間モデル (Dynamic Linear Model : DLM) [46] [47] [48] [49] [50] [51]などがあげられる。Trust-Distrust モデルでは、メディア変数とエージェント間のインタラクションから、イノベーションに対する態度のシミュレーションを行っている。それに対し VAR と DLM では、ともに独立変数に設定した広告やプロモーションといった自社でコントロール可能な変数などから、従属変数に設定した売上など推定するモデルとなっている。

自己回帰 (AR) モデルをベクトルに一般化したものが VAR であり [46] [47]、VAR(p)モデルは y_t を定数と自身のp期の過去の値に回帰したモデルである。

$$y_t = c + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} + \varepsilon_t \quad (8)$$

- y_t はn個の変数ベクトル,
- c は $n \times 1$ の定数ベクトル,
- ϕ_i は $n \times n$ の係数行列

n 変量VAR(p)モデルは n 本の回帰式からなり、それぞれの回帰式は各変数を定数と全変数の p 期間の過去の値に回帰した形となる。また全ての回帰式が同一の説明変数をもつ2変量のVAR(1)モデルは以下となる。

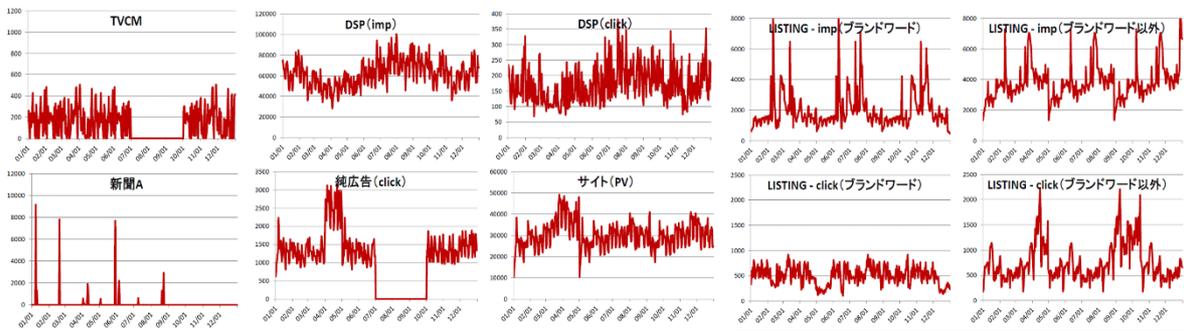
$$y_t = c_1 + \phi_{11}y_{t-1} + \phi_{12}x_{t-1} + \varepsilon_{1t} \quad (9)$$

$$x_t = c_2 + \phi_{21}y_{t-1} + \phi_{22}x_{t-1} + \varepsilon_{2t} \quad (10)$$

同時点の相手のデータはモデルに含まれない
ただし(9)(10)式の y_t などは、(8)式のベクトル y_t のうちの成分

VAR モデルを用いる目的は、モデルの精度向上のために複数の変数を用いること、そして変数間のダイナミックな関係性の分析を行うことにある。特にデータ間の因果関係の有無の検定する Granger の因果性、変数 x_t (広告) が増加すると変数 y_t (売上) にはどういった影響があるのかを定量的に評価するインパルス応答関数、そして各変数の不確実性 (予測できない変動) において他の変数が寄与する割合を計算する分散分解というツールが利用できるメリットは大きい [46] [47]。図 7 では、VAR モデルを構築する際に使用したデータイメージを示している。分析対象期間の売上データと当該期間に投下された複数の広告やキャンペーンデータの時系列グラフである。

コミュニケーション施策やその他の要因



目的変数
(売上等)



図 7：VAR モデルで使用するデータ

また図 8 では、これらのデータを利用して構築された VAR モデルの結果の一例を提示している。売上に対する各種広告効果の積み上げグラフとモデルの予測値の折れ線グラフである。このように売上高に対する各広告施策の貢献度を見える化することで広告の効果検証の他、マーケティング・コミュニケーション施策の立案にも役立てられている。

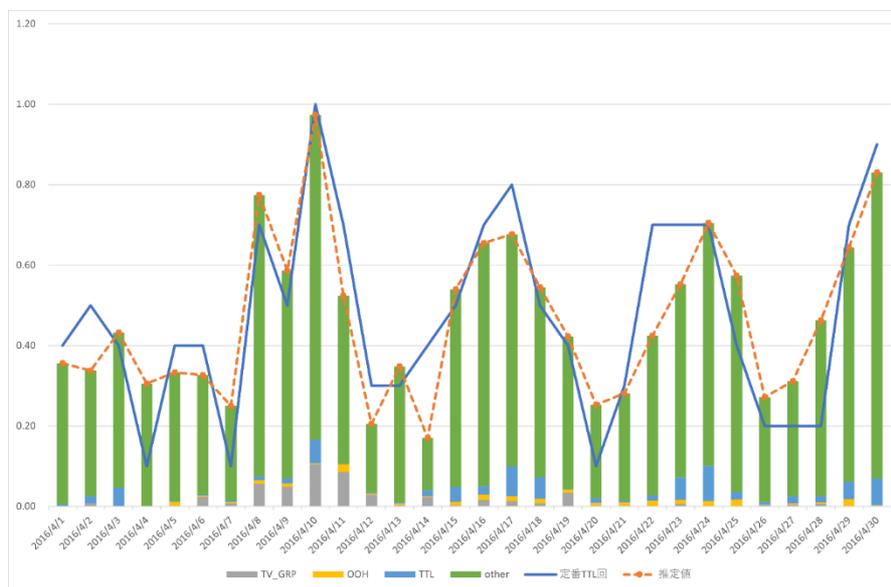


図 8：VAR モデルアウトプット例_広告効果の積み上げグラフ

状態空間モデルは、目に見えない状態の変化と、観測値を取り扱う時系列モデルの一つである。図9で示したように、状態の変化を表す状態（システム）モデルと観測値が得られるプロセスを表す観測モデルで構成される。

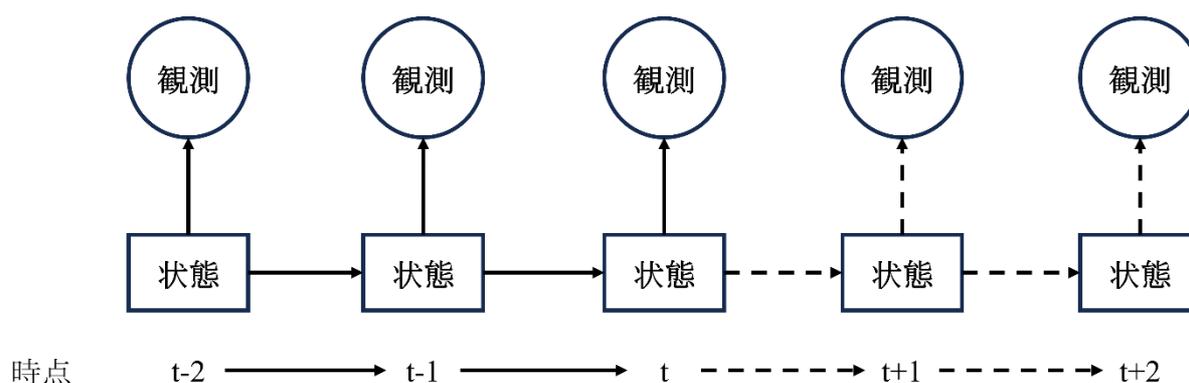


図9：状態空間モデルのグラフィカルモデル

時系列データ $y_{1:n}$ の挙動を説明する基本的な状態空間モデルは以下の2本の式で構成される。

$$y_t = Z_t \alpha_t + \epsilon_t \quad (11)$$

$$\alpha_{t+1} = T_t \alpha_t + \eta_t \quad (12)$$

式(11)は観測変数方程式、(12)は状態方程式と呼ばれる。状態空間モデル内の α_t は状態変数と呼ばれ観測されない確率変数である。観測される y_t の背後に存在する隠れた構造を表している。状態空間 α_t と y_t の時系列構造は図10にまとめられる。また各種広告施策を独立変数とし、ブランド認知率を従属変数とした、状態空間モデルによる計算結果の例を図11に示している。ここでは青線が実際のブランド認知率を示しており、オレンジ線が予測値となっている。実測値のピーキーな時点では実測値と予測値が乖離している個所もあるが、トレンドをトレースしていることが確認できる。

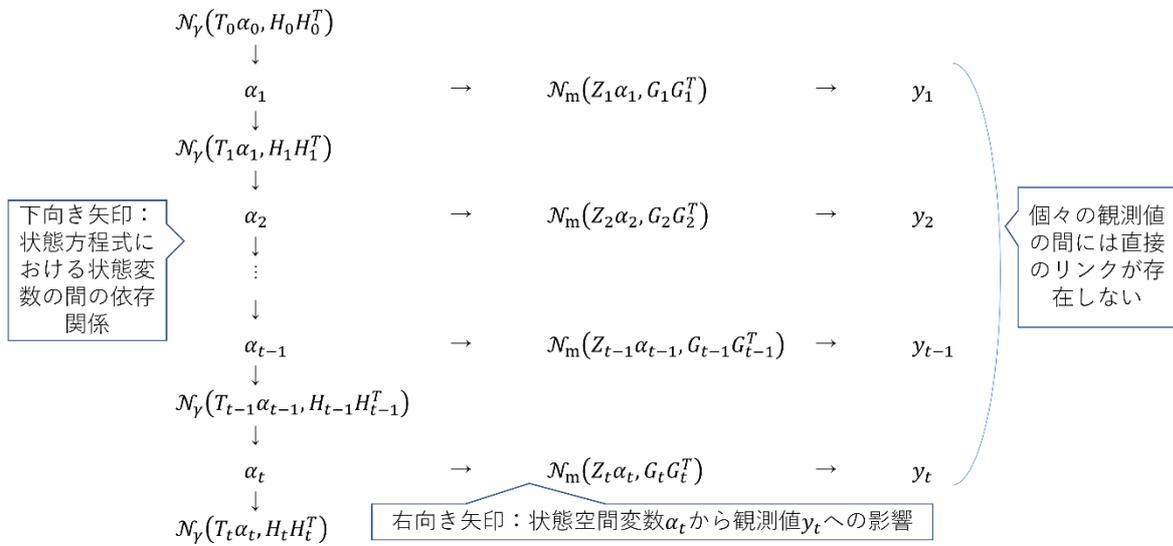


図 10：状態空間モデルの様式図

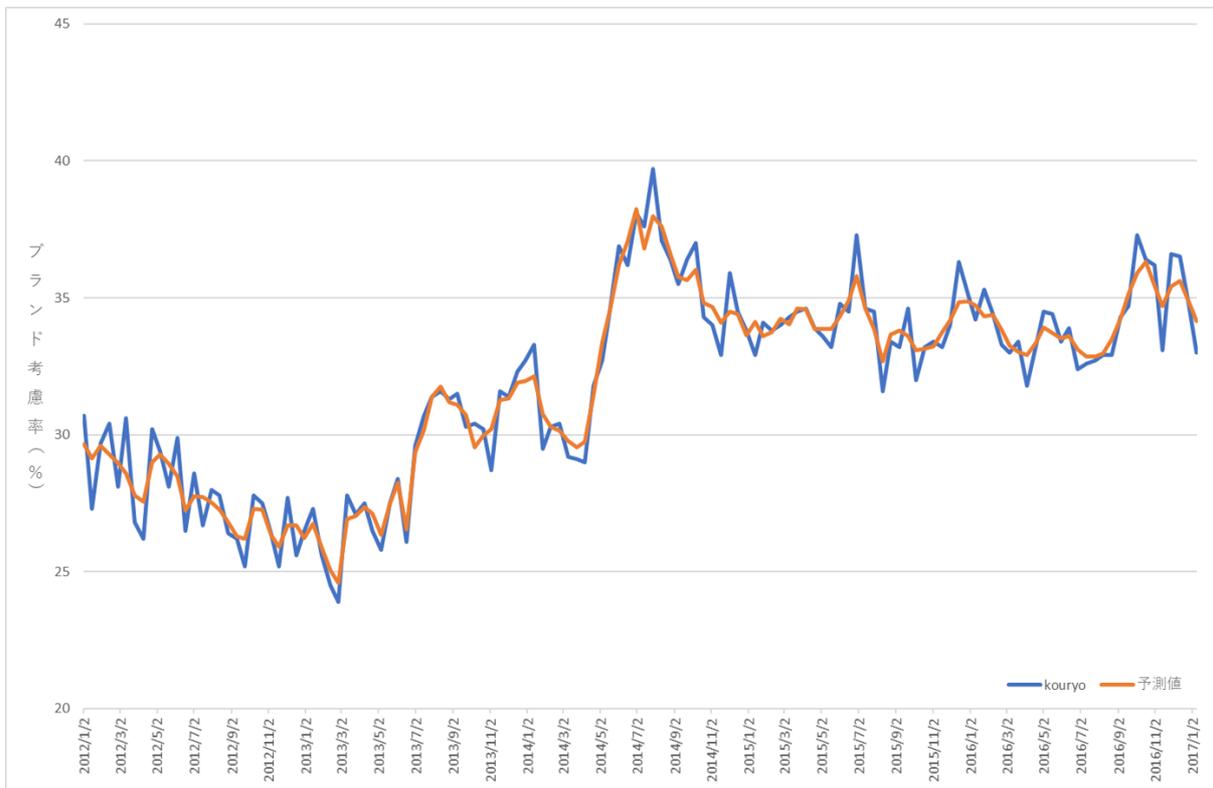


図 11：状態空間モデルによる計算結果の例

第3章 エージェントと普及/伝播の対象に関する議論

2.4 節で述べたように、通常の Trust-Distrust モデルでは、エージェントは1つのカテゴリとして扱われ、シミュレーションのスタート時から全員が一斉にシミュレーションの対象となる [31] [32]。本稿ではイノベーションを採用する採用者カテゴリの時差も考慮するため、エージェントを分割し時差をもってシミュレーションに投下することで、Trust-Distrust モデルにおけるイノベーションに対するエージェントの態度が形成される推移を考察する。

エージェントのカテゴリ数を変更する理由としては、消費者の革新性の違いが挙げられる。エージェントを革新性を軸としてセグメント化しシミュレーションを実行することで、エージェントの革新性の違いによる態度形成が見える化されることのメリットは大きいといえる、

はじめのシミュレーションでは、エージェントを2カテゴリ：イニシャルユーザー N_{ini} と、ニューユーザー N_{user} に分けてシミュレーションを実行する。後半のシミュレーションでは、エージェントを5つの採用者カテゴリに分類し、シミュレーションを実行する。

最初の計算例として、ここでは $A(t)$ を定数としてシミュレーションを実行する。図 11 は、石井らのオピニオンダイナミクス理論を用いて、エージェント数 300 で計算された例を示している。人々のネットワークは、リンクの接続確率が 50% であるランダムネットワークである。エージェント間の相互 D_{ij} 値は、-1 から 1 の間の均一乱数によって決定される。またシミュレーションは 10 ステップである。図からわかるように、人々の態度はポジティブとネガティブに散らばっている。この計算ではマスメディア効果 $A(t)$ を 0 と設定しているため、ポジティブ方向へ偏ることなく態度の分布は正負に散らばっている。本稿では、エージェント間の相互作用およびメディア接触による正または負への態度変容に注目するので、態度分布の時間的发展及び正負の比率に着目することとする。

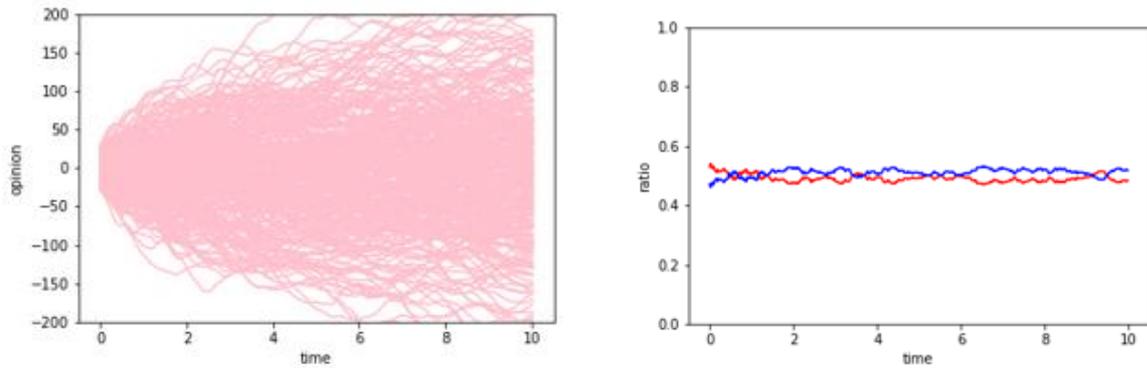


図 12 : $N = 300$ の計算結果. 人的ネットワークは, リンク接続確率 0.5 のランダムネットワーク. 左図は態度の軌跡の時間発展. 右図は正(+)と負(-)の比率の時系列変化, 赤が正(+), 青が負(-). D_{ij} はランダムに -1 から 1 に設定. マスメディア効果はゼロ ($A(t) = 0$).

3.1 2 カテゴリーエージェントによるシミュレーション

本節では石井らのモデル [31] [32] をベースとし, ここに新規参入者を加えることでシミュレーションを実行する. 本節のシミュレーションでは, 社会全体を 300 人 ($N = 300$) として, 最初から市場に存在する消費者を N_{ini} , あとから市場に参入してくる消費者を N_{user} としている. つまり $N = N_{ini} + N_{user} = 300$ となる. また消費者の当該市場への後からの参入タイミングについては以下のように定義し, 順次市場に参入してくると仮定している.

$$N_{user} = \frac{(N - N_{ini})}{2} * (\tanh x - 1) \quad (13)$$

3.1.1 最初から興味をもって市場に存在する消費者の操作

最初から興味をもって市場にいる消費者の人数と途中から興味をもって市場に参加する消費者, マスメディア係数, そしてランダムネットワークのリンク接続確率の操作を行い, 途中から興味をもって市場に参入してくる消費者の態度形成の変遷を計算する.

最初から興味をもって市場に存在する消費者の操作を行う。 $N = 300$ とするが、最初に興味をもって市場に存在する消費者 (N_{ini}) を 0, 1, 5, 10, 50, 100, 200 と設定する。例えば $N_{ini} = 1$ の場合、当該市場には一人の消費者が存在し、299 人の消費者が当該市場に後から興味を示して参入してくるということである。ここでも $A(t)$ が定数であることを仮定しており、人々のネットワークも、リンクの接続確率が 50% であるランダムネットワークだと仮定している。

図 13 で $N_{ini} = 0$ の場合のシミュレーションの結果を示している。 $N_{ini} = 0$ ということは現実の状況では考えにくいですが、事前に興味をもった消費者は市場に存在せず、全員が後から当該イノベーションに興味をもち参入してくる、ということである。

当該イノベーションにニュートラルな立場で参入してくる消費者は、マスメディアやイノベーションに対して正 (+) または負 (-) の態度を持った周囲の人々の影響を受けて自身の態度を変容させていく。しかしながら今回の計算結果では、式(5)右辺最後の項で D_{ij} はゼロではないが $I_i = I_j = 0$ であるため、 $\Delta I_i = 0$ となり、態度変容は見られず終始 0 に留まるという結果となった。つまりイノベーションの採用を後押しするためには、イノベーションに対して好意的な評価を持った消費者が事前に存在するか、態度変容を促す広告などに接触することが必要があると考えられる。マスメディアの効果については次節で検証する。

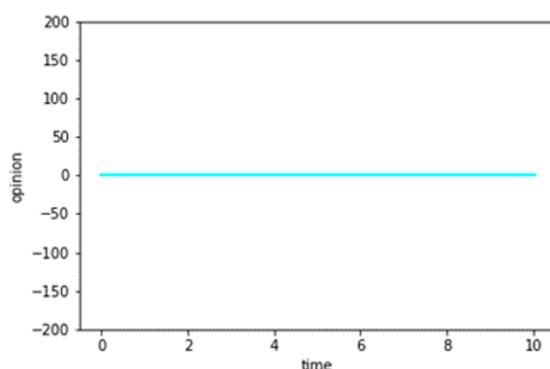


図 13 : $N = 300$ の計算結果。事前に存在する N_{ini} は 0 人。人的ネットワークは、リンク接続確率 0.5 のランダムネットワーク。図は態度の軌跡の時間発展。 D_{ij} はランダムに -1 から 1 に設定。マスメディア効果はゼロ ($A(t) = 0$) 。

次いで N_{ini} が 1, 5, 10 の場合のシミュレーション結果である(図 13). 事前に興味を持った人の態度の初期分布は ± 30 の範囲で設定されている. 300 人の間の相互 D_{ij} 値は, -1 から 1 の間の一様乱数によって決定される. 図からわかるように, $N_{ini} = 0$ の時とは異なり, 人々の態度はポジティブとネガティブに散らばっている. この計算でもマスメディア効果 $A(t)$ を 0 と設定しているため, この図の態度分布はバイアスがなくて正と負にほぼ同数に分布する.

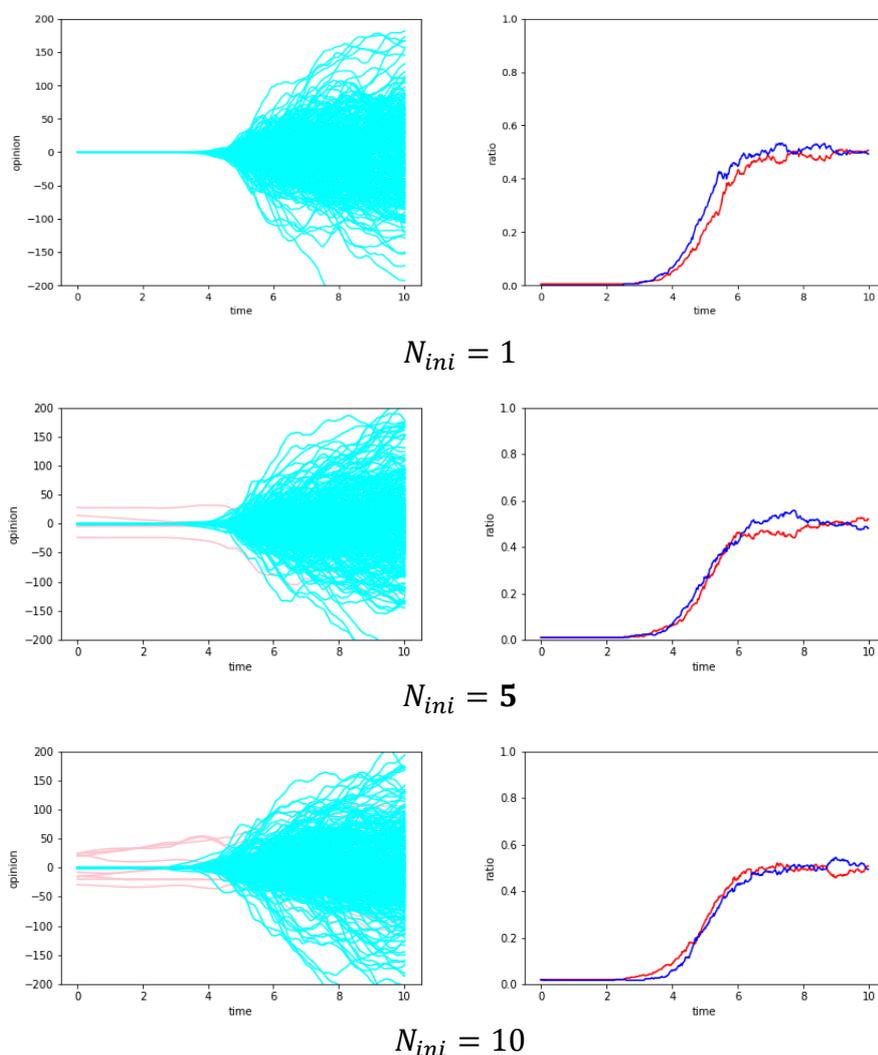
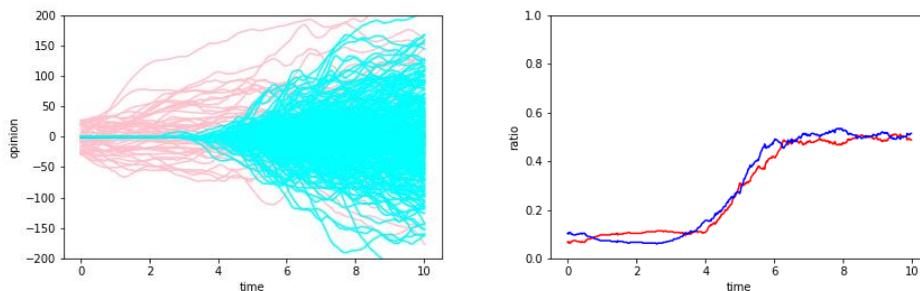
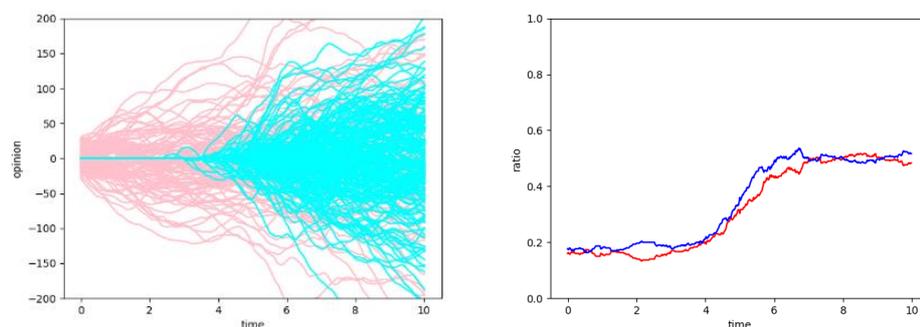


図 14 : $N = 300$ の計算結果. 事前に存在する N_{ini} は上より 1, 5, 10 人. 人的ネットワークは, リンク接続確率 0.5 のランダムネットワーク. 左図は態度の軌跡の時間発展であり, 赤が N_{ini} , 青が N_{user} を示している. 右図は正(+)と負(-) の比率の時系列変化, 赤が正(+), 青が負(-). D_{ij} はランダムに -1 から 1 に設定. マスメディア効果はゼロ ($A(t) = 0$).

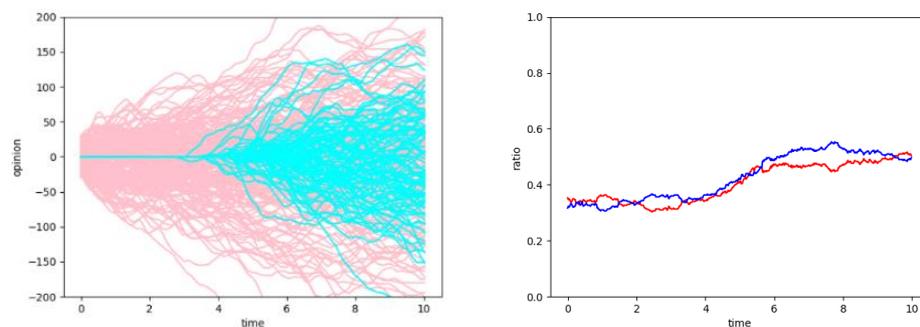
図 15 では N_{ini} が 50, 100, 200 について同様の計算を示している. 図 14 と 15 の計算結果を見ると, 最初から興味をもって市場に存在する消費者の大小は態度形成に関係ないように見て取れる. これはおそらく人の態度は正 (+) と負 (-) の相対的なボリュームによって左右されるのではなく, N_{ini} の意見分布が $-1 \sim +1$ の一様乱数であることが前提であるが, エージェント間の相互作用によって態度が導かれるからである.



$N_{ini} = 50$



$N_{ini} = 100$



$N_{ini} = 200$

図 15 : $N = 300$ の計算結果. 事前に存在する N_{ini} は左より 50, 100, 200 人. 人的ネットワークは, リンク接続確率 0.5 のランダムネットワーク. 図は態度の軌跡の時間発展であり, 赤が N_{ini} , 青が N_{user} を示している. D_{ij} はランダムに -1 から 1 に設定. マスメディア効果はゼロ

$$(A(t) = 0) .$$

3.1.2 マスメディア効果の操作

次いで、一定のメディア効果をオピニオンダイナミクス理論に適用し、その影響が新規参入者にどのような影響を与えるのか考察する。すなわち、 $A(t) = A_c$ であると想定する。ここで A_c は定数値である。図 16 は図 15 と比較して、広告効果 $A(t)$ だけが違っている。ここで図 16 では、 $A(t)$ は 1, 5, 10 であると仮定している。

$A(t) = 1$ と5を比較して、 $A(t) = 5$ の態度分布は、正の態度に向かってバイアスがかかっているように見える。 $A(t) = 10$ のケースでは、計算された態度分布は明らかに正の態度方向に歪んでいる。これは石井らのオピニオンダイナミクス理論において、メディア効果が市場の態度をメディア主導の方向に偏らせる現象を、定性的に説明していることがみてとれる。その効果は、既存の消費者だけではなく、後から参入してくる消費者にも同様に示している。また先に市場に存在しより長くマスメディアに接触している人の方が、より強くメディアの効果を受けるように見て取れる。その結果、より多くの正態度の人と接触することで、後から参加してくる消費者の態度に影響を与えると考えられる。

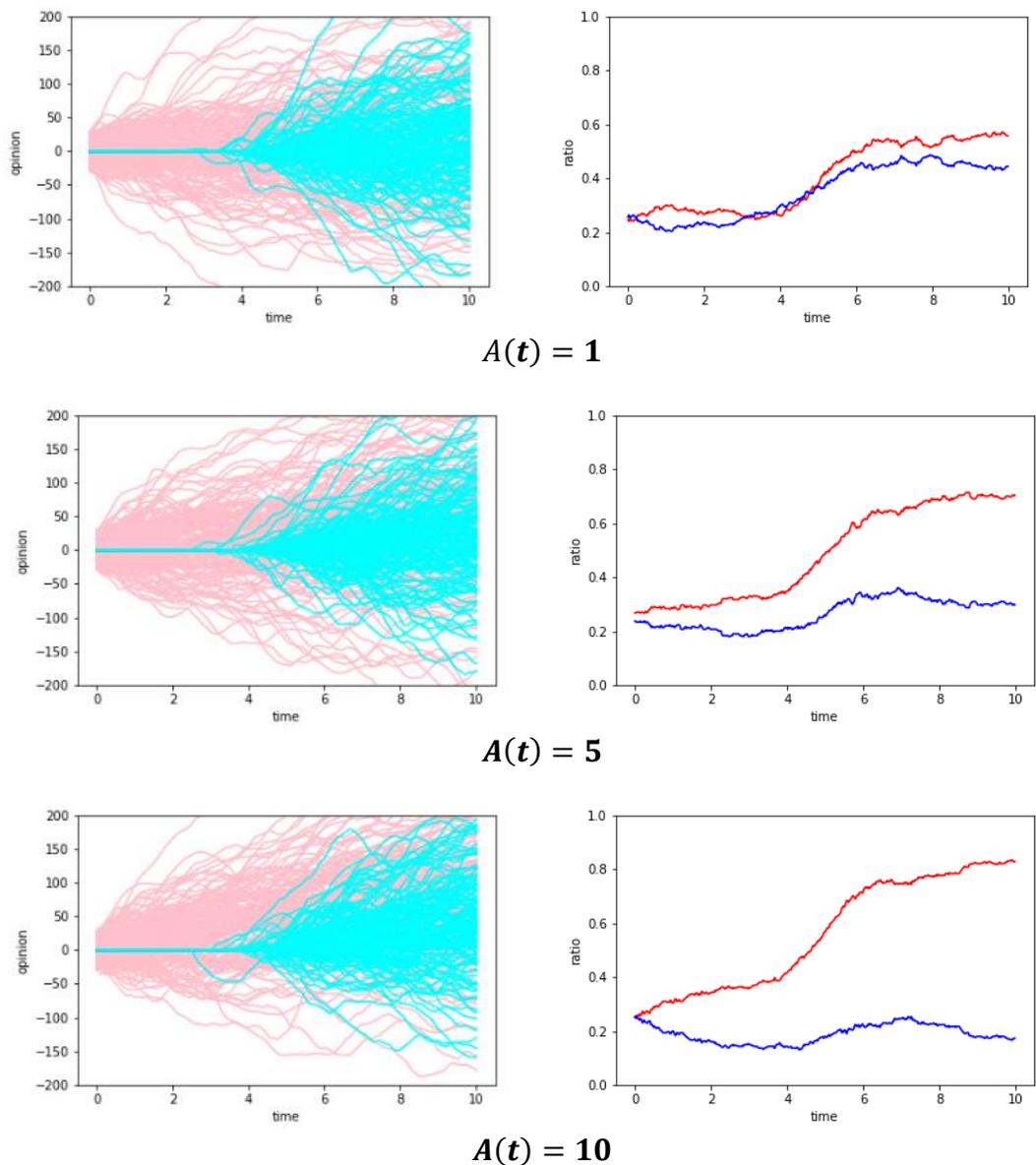


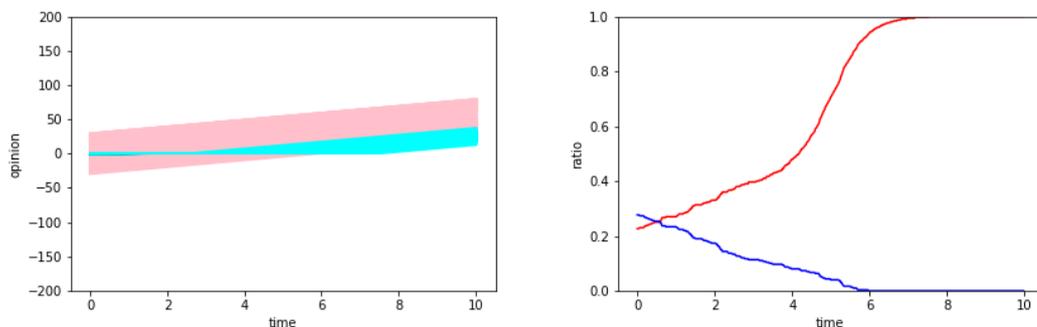
図 16 : $N = 300$ の計算結果. 事前に存在する N_{ini} は 150 人. 人的ネットワークは, リンク接続確率 0.5 のランダムネットワーク. 左図は態度の軌跡の時間発展であり, 赤が N_{ini} , 青が N_{user} を示している. 右図は正(+)と負(-)の比率の時系列変化であり, 赤が正(+), 青が負(-). D_{ij} はランダムに -1 から 1 に設定. マスメディア効果は 1, 5, 10 ($A(t) = 1, 5, 10$) と設定.

3.1.3 ランダムネットワークの操作によるシミュレーション

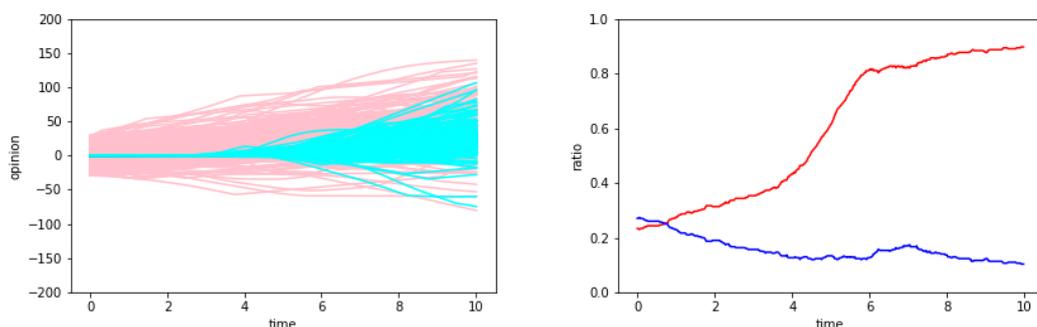
ついでランダムネットワークのノード間の接続確率を変えながら計算を行う. 接続確率の高低は, 実社会における人付き合いの濃淡を想定している. 人付き合いの濃い

ネットワークとしては、多くの住民同士が顔なじみといった村落や特定ブランドのコミュニティがあげられる。また人付き合いの淡いネットワークとしては、隣人の顔や名前を知らないといった都市部の集合住宅などがあげられる。他者との結びつきの高低が態度変容に影響を与えるのかを検証する。計算では $N = 300$ に設定 ($N_{ini} = 150$)。 D_{ij} はランダムに-1 から 1 に設定されているため、集団の中でエージェントの信頼関係は偏っていない。ここでマスメディアの影響を $A(t) = 5$ に設定する。ランダムネットワークで他のノードに接続する確率を 0%, 1%に変更する。計算結果を図 17 に示す。

この計算から、ノード間の接続の確率が 0%の場合では、消費者は他者の影響を受けず、マスメディアにのみ影響を受け態度を形成していることが見て取れる。当初マイナスに位置していた消費者も時間の経過によって、マイナスからプラスへ態度が移動していることが見て取れる。接続確率が 1%の場合も分布は正寄りとなつてはいるが、態度分布はマイナスにまで広がっていることが見て取れる。これは一方的にマスメディアの影響を受けるだけではなく、他人の態度の影響も受けた結果、態度の分布が広がっていることを意味する。



ランダムネットワークの接続確率：0%

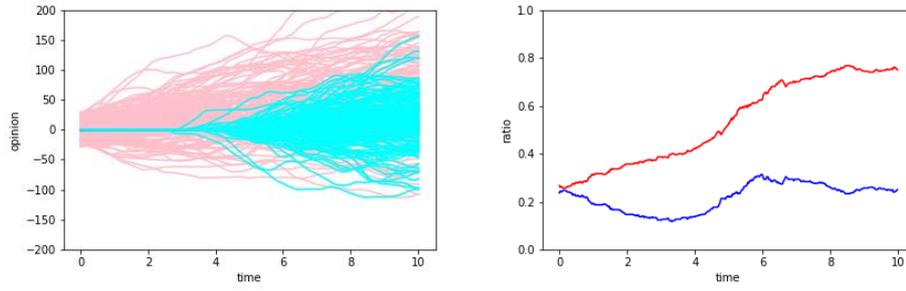


ランダムネットワークの接続確率：1%

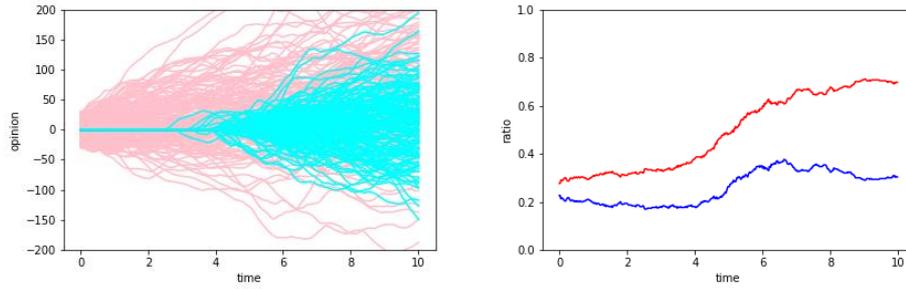
図 17: $N = 300$ の計算結果. 事前に存在する N_{ini} は 150 人. 左図は態度の軌跡の時間発展であり, 赤が N_{ini} , 青が N_{user} を示している. 右図は正(+)と負(-)の比率の時系列変化, 赤が正(+), 青が負(-). D_{ij} はランダムに-1 から 1 に設定. マスメディア効果は 5 ($A(t) = 5$). 人的ネットワークは, リンク接続確率 0%, 1%のランダムネットワーク.

ランダムネットワークの接続確率が 0%と 1%で態度分布の違いを確認したが, さらにランダムネットワークで他のノードに接続する確率を 10%, 30%, 50%そして 90%に変更する. 計算結果を図 18 に示す.

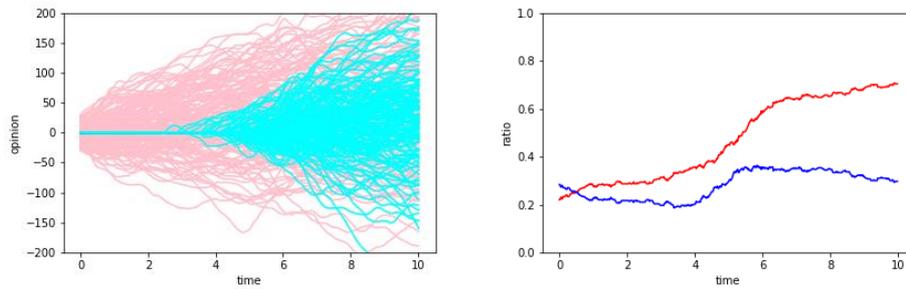
この計算からノード間の接続の確率が上がるほど, 人の態度は他人の態度の影響を受けることが分かる. 10%と 90%の態度の散らばりを見ても, 明らかに 90%の方が分布が広がっていることが見て取れる. つまり人と人の繋がりが疎の場合には, マスメディアの影響をより受けて態度が形成されるが, 接続確率が高い場合, 他人の態度の影響を大きく受けメディアが効きにくいことが示唆される.



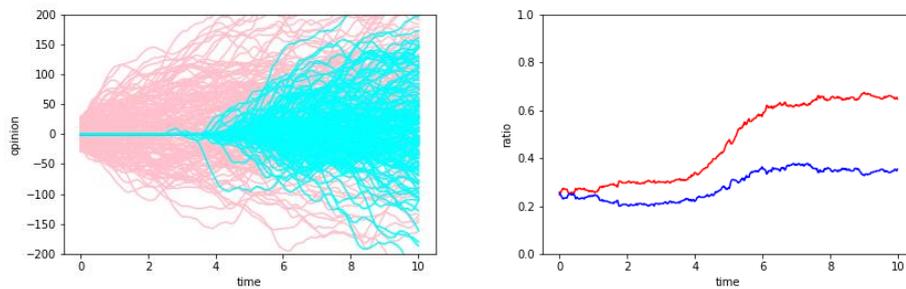
ランダムネットワークの接続確率：10%



ランダムネットワークの接続確率：30%



ランダムネットワークの接続確率：50%



ランダムネットワークの接続確率：90%

図 18 : $N = 300$ の計算結果. 事前に存在する N_{ini} は 150 人. 左図は態度の軌跡の時間発展であり, 赤が N_{ini} , 青が N_{user} を示している. 右図は正 (+) と負 (-) の比率の時系列変化, 赤が正 (+), 青が負 (-). D_{ij} はランダムに -1 から 1 に設定. マスメディア効果は 5 ($A(t) = 5$) と設定. 人的ネットワークは, リンク接続確率 10%, 30%, 50%, 90% のランダムネットワーク.

3.2 5 カテゴリーへエージェントの拡張

前節では、エージェントを N_{ini} と N_{user} に分けてシミュレーションを実行したが、本節では 2.5 節で記述したロジャースの5つの採用者カテゴリーにエージェントを拡張し、イノベーションに対する態度の推移を考察する。採用者カテゴリーの構成については、イノベーター2.5%、アーリーアダプター13.5%、アーリーマジョリティ 34%、レイトマジョリティ 34%、ラガード 16%に設定している。

エージェントのシミュレーションへの投下については、本研究でも藤居/石井 [52]と同じ関数を採用するが、各採用者カテゴリーがシミュレーションに参加するタイミングについては以下の通り時間差を考慮している(図 19)。また特段の説明がない場合、各カテゴリーの意見は市場に投下されたタイミングでは0である。

$$\text{イノベーター} : N_{inn} = \frac{(\tanh(x) + 1)}{2} \quad (14)$$

$$\text{アーリーアダプター} : N_{ea} = \frac{(\tanh(x - 3) + 1)}{2} \quad (15)$$

$$\text{アーリーマジョリティ} : N_{em} = \frac{(\tanh(x - 6) + 1)}{2} \quad (16)$$

$$\text{レイトマジョリティ} : N_{im} = \frac{(\tanh(x - 9) + 1)}{2} \quad (17)$$

$$\text{ラガード} : N_{lg} = \frac{(\tanh(x - 12) + 1)}{2} \quad (18)$$

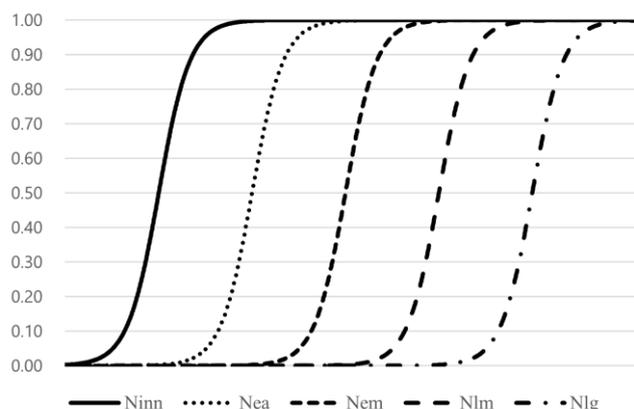


図 19 : 各採用者カテゴリーのイノベーション採用タイミング。

石井 [32]モデルをベースに，エージェントを5つの採用者カテゴリーに分類し，態度の初期分布，信頼係数，メディア係数の接続確率の操作を行い，イノベーションの伝播の変遷を計算する．

3.2.1 5 エージェントモデルによるシミュレーション

最初の計算例として，1000人15ステップで計算された結果を図20に示している．イノベーターの態度の初期値は-30から+30の間に一様乱数によって分布，人々のネットワークはリンク接続確率が50%のランダムネットワークと設定している．1000人の間の D_{ij} 値は，-1~1の一様乱数により決定される．またマスメディア効果 $A(t)$ を0と設定して計算しているため，図からわかるように，人々の態度はバイアスがなく正と負にほぼ均等に分布している．図20の右図はエージェントのイノベーションに対する態度の正負それぞれの比率の時系列変化を示したものである．ロジャースは「イノベーションを採用する人数を時間軸に沿って累積的に図に描くと，その分布はS字型の曲線となる」(Rogers, 2003 三藤訳, 2020, p31)と述べている．シミュレーションの結果も正(または負)の態度を示すエージェントの数を時間軸に沿って累積するとS字型の曲線を描いていることが確認できる．本稿ではロジャースの普及曲線になぞらえて，それぞれを採用曲線と拒絶曲線とここでは呼ぶこととする．本章では，この計算結果をベースにシミュレーションを実行する．

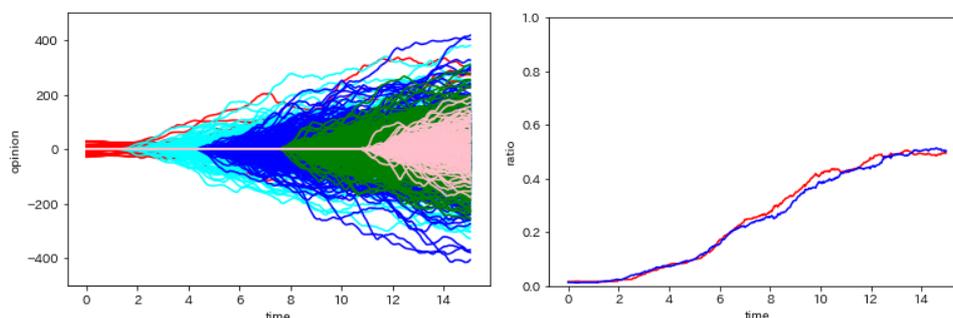


図 20 : $N = 1000$ の計算結果．人的ネットワークはリンク接続確率0.5のランダムネットワーク．左図は各採用者カテゴリーの態度軌跡の時間発展(赤： N_{inn} ，水色： N_{ea} ，青： N_{em} ，緑： N_{lm} ，ピンク： N_{ig})．右図は正(+)と負(-)の比率の時系列変化． D_{ij} はランダムに-1から1に設定．メディア効果はゼロ ($A(t) = 0$) ．

3.2.2 態度の初期分布を操作したシミュレーション

図 20 のシミュレーションではイノベーターの初期の態度分布を -30 から $+30$ の間に一様乱数によって設定したが、ここでは $10\sim 30$ あるいは $-10\sim -30$ とバイアスをかけ、イノベーターの状態が後からシミュレーションに参加するエージェントにどのような影響を与えるのか、態度の伝播を観察する。図 21 の左図はプラス、右図はマイナスからイノベーターの態度分布をスタートさせているが、正と負のシェアは拮抗しているようにみとれる。これは信頼係数 D_{ij} がランダム($-1\sim 1$)に設定されている上、プラスまたはマイナスに偏らせるバイアス変数が態度の初期値以外に存在しないため、人々がインタラクションしている過程で均一化されていくからだと考えられる。

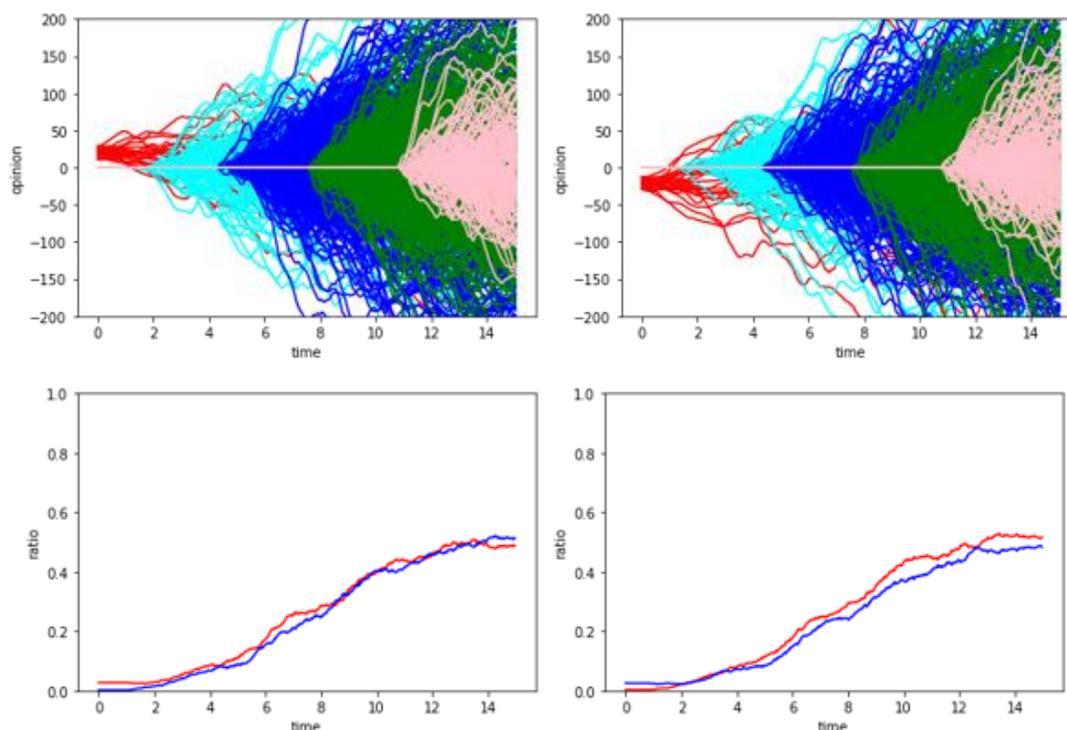


図 21 : $N = 1000$ の計算結果. 左図 : イノベーターの初期の態度分布 = プラス ($10\sim 30$) , 右図 : イノベーターの初期の態度分布 = マイナス ($-10\sim -30$) . 各採用者カテゴリーの態度軌跡の時間発展 (赤 : N_{inn} , 水色 : N_{ea} , 青 : N_{em} , 緑 : N_{lm} , ピンク : N_{lg}). 人的ネットワークはリンク接続確率 0.5 のランダムネットワーク. D_{ij} はランダムに -1 から 1 に設定. メディア効果はゼ

$$\square (A(t) = 0) .$$

また採用者カテゴリーのうち、アーリーアダプターが普及の鍵を握るとされているため、アーリーアダプターの態度分布も偏らせ計算した結果が図 22 である。

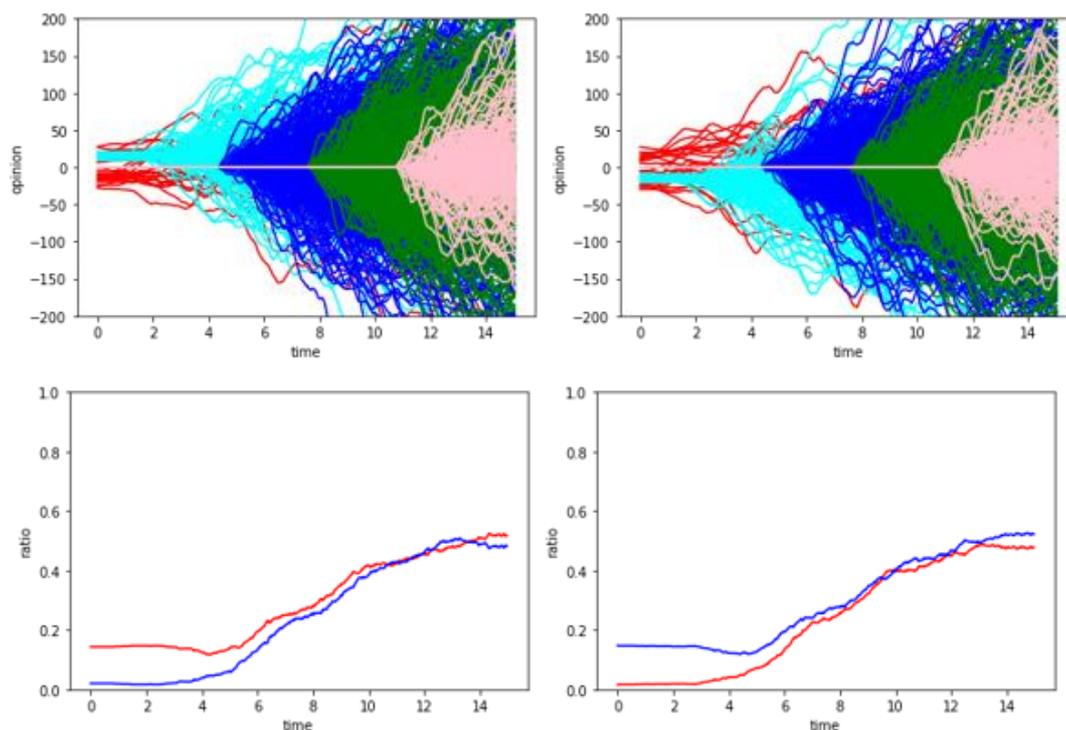


図 22 : $N = 1000$ の計算結果. 左図 : アーリーアダプターの初期の態度分布 = プラス (10 ~ 30) , 右図 : アーリーアダプターの初期の態度分布 = マイナス (-10 ~ -30) . 各採用者カテゴリーの態度軌跡の時間発展 (赤 : N_{inn} , 水色 : N_{ea} , 青 : N_{em} , 緑 : N_{lm} , ピンク : N_{lg}). イノベーターの初期の態度分布 ± 30 . 人的ネットワークはリンク接続確率 0.5 のランダムネットワーク .

D_{ij} はランダムに -1 から 1 に設定 . メディア効果はゼロ ($A(t) = 0$) .

アーリーアダプターの態度をプラスからスタートさせたシミュレーションの前半ではプラスの比率が高く、マイナスの場合はマイナスの比率が高いが、シミュレーション後半ではプラス/マイナスの比率は同程度という結果を得た。こちらも信頼係数 D_{ij} がランダム (-1 ~ 1) に設定されている上、プラスまたはマイナスに偏らせるバイアス変数が態度の初期値以外に存在しないため、人々の意見交換といった相互作用がなされている過程で、態度が平準化されていくものと考えられる。

3.2.3 信頼係数 D_{ij} の操作

ではアーリーアダプターが他の採用者カテゴリーから信頼されていたらどうであろうか. ここでは基本条件は図 22 と同じとしつつも, アーリーアダプター, アーリーマジョリティ, レイトマジョリティ, ラガードからアーリーアダプターへの信頼係数を 1~2(非負)と設定し, シミュレーションを実行する(図 23). アーリーアダプターの初期の態度分布がプラスの場合はプラス, マイナスの場合はマイナスの比率が上をいっていることが確認できる. これはアーリーアダプターがイノベーター以外の採用者カテゴリーから信頼されているので, アーリーアダプターの正の態度が他 3 カテゴリーに影響を与えていると考えられる. しかしながらアーリーアダプターが信頼されているという状態は, イノベーションを普及させたい側からすると, 毒にも薬にもなるということを示唆しており, アーリーアダプターを味方につけておくための施策(態度を正方向に歪める変数)が必要ということになる.

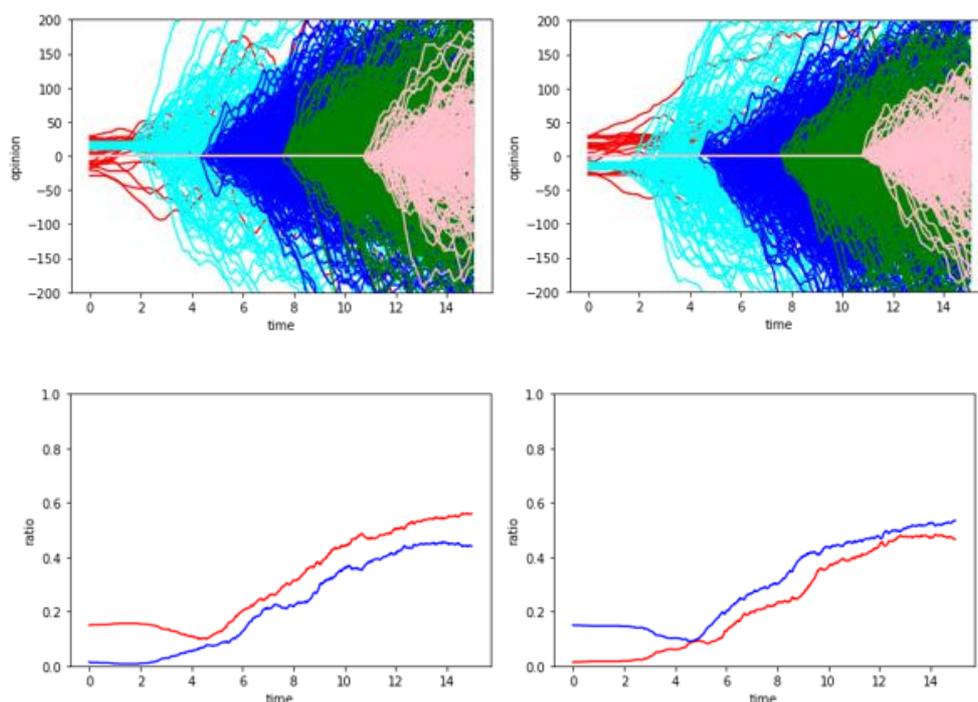


図 23 : $N = 1000$ の計算結果. 左図 : アーリーアダプターの初期の態度分布=プラス (10~30), 右図 : アーリーアダプターの初期の態度分布=マイナス (-10~-30). 各採用者カテゴリーの態度軌跡の時間発展(赤 : N_{inn} , 水色 : N_{ea} , 青 : N_{em} , 緑 : N_{lm} , ピンク : N_{lg}). イノベーターの初期の態度分布 ± 30 . 人的ネットワークはリンク接続確率 0.5 のランダムネットワーク. アーリーアダプター, アーリーマジョリティ, レイトマジョリティ, ラガードからアーリーアダプターへの D_{ij} はランダムに 1~2, それ以外の D_{ij} はランダムに-1~1 に設定. メディア効果はゼロ ($A(t) = 0$).

3.2.4 マスメディア効果の操作

一定のマスメディア効果をオピニオンダイナミクス理論に適用し、その影響が各採用者カテゴリーと正と負の比率にどのような影響を与えるのか考察する。すなわち、 $A(t) = A_c$ であると想定する。ここで A_c は定数値である。図 24 は図 20 と同じ計算条件で計算したものだが、メディア効果 $A(t)$ はゼロではない。図 24 では、 $A(t) = 2.5, 5, 10$ と設定している。 $A(t) = 2.5$ と 5 を比較して、 $A(t) = 5$ の態度分布は、正の態度に向かってよりバイアスがかかっているように見える。

$A(t) = 10$ のケースでは、計算された態度分布は明らかに正方向に歪んでいる。これは石井らのオピニオンダイナミクス理論において、メディア効果が市場の態度をメディア主導の方向に偏らせる現象を、定性的に説明していることがみてとれる [31] [32]。その効果はイノベーターだけではなく、後から参入してくる採用者カテゴリーにも同様の効果を示している。また先に市場に存在し、より長くマスメディアに接触している採用者カテゴリーの方が、より強くメディアの効果を受けるように見て取れる。その結果、より多くの正の態度をもつ人と接触することで、後から参加してくる採用者カテゴリーの態度に影響を与えたと考えられる。

他のシミュレーション条件が同じである場合、より大きなマスメディア変数を投下することによって、メディアが主導したい方向に採用者カテゴリーの態度を歪ませることが出来ることを示唆している。広告といった刺激を消費者に感知させ態度変容を起こさせるには、ある程度の広告量が必要だとされ、それは閾値と呼ばれている [53]。今回のシミュレーションでは、 $A(t) = 2.5, 5, 10$ の比率(図 24 下段の折れ線グラフ)を比較した場合、広告量に応じて線型に正の比率が大きくなっていることは確認できるが、閾値の存在を示唆する結果は得られなかった。閾値の検証については、シミュレーションの拡充の他、実データを参照するなどしたモデルの改良が必要であると思われる。

またその一方でマスメディア変数の投下によって正方向にバイアスの掛かるシミュレーションにおいても、一定数の負の態度を示すエージェントが存在するのも事実である。これはエージェント間のインタラクションが強力であることを示していると考えられる。

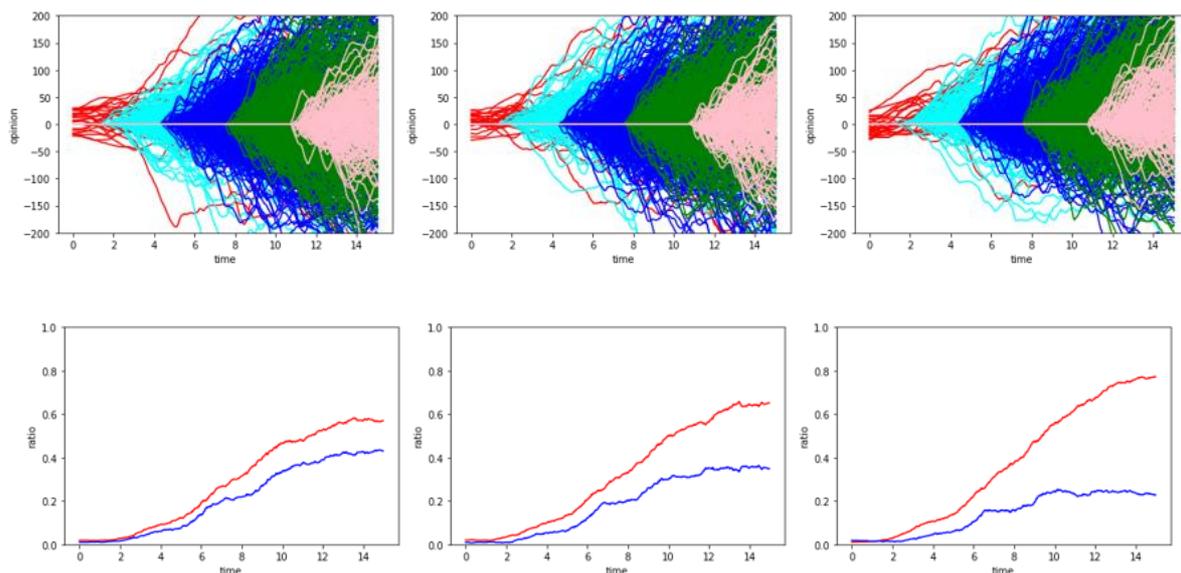


図 24 : $N = 1000$ の計算結果. 人的ネットワークはリンク接続確率 0.5 のランダムネットワーク. 上図の 3 つは各採用者カテゴリーの態度軌跡の時間発展(赤 : N_{inn} , 水色 : N_{ea} , 青 : N_{em} , 緑 : N_{lm} , ピンク : N_{lg}). 下折れ線グラフは正 (+) と負 (-) の比率の時系列変化. D_{ij} はランダムに -1 から 1 に設定. マスメディア効果は左より $A(t) = 2.5, 5, 10$.

ついでアーリーアダプターの信頼係数を非負の設定とした上で、マスメディア変数を操作したシミュレーションを実行する。イノベーション伝播の鍵となるアーリーアダプターが当該イノベーションの採用にプラスで、イノベーターを除く採用者カテゴリーがアーリーアダプターを信頼している環境下で、正方向のマスメディア変数を設定した場合のシミュレーションとなる(図 25)。

マスメディア変数が投下されているので、左図下の時系列グラフと比較して、右図下の時系列グラフの方が正にバイアスがかかっていることが確認できる。また正と負の比率の時系列グラフでは、シミュレーション前半の動きは左右ともに同じような動きをしているが、後半では負の態度方向が失速し、正の比率が伸びていることが確認

できる. オピニオンリーダーと目されるアーリーアダプターがイノベーションの採用に積極的な意向を持ち, 他の採用者カテゴリーから信頼を得ており, メディアが正の態度に投下される好環境が構築できれば, 多数派の取り込みが期待できることを示唆した結果といえる.

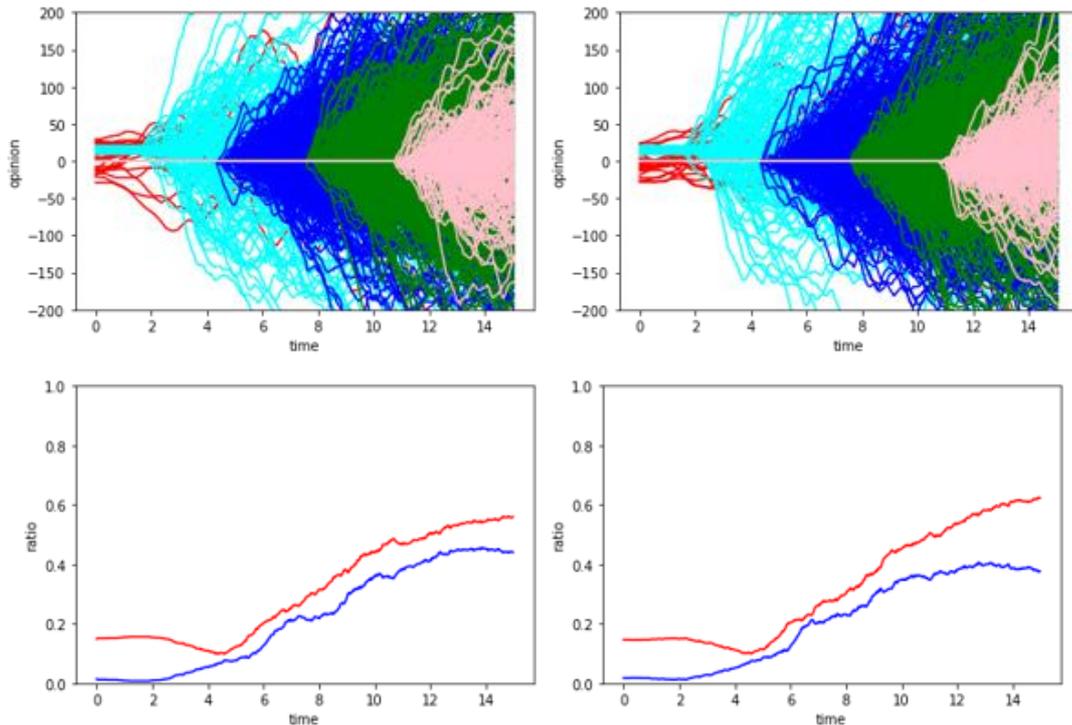


図 25 : $N = 1000$ の計算結果. 左右図とも, イノベーターの初期の態度分布 ± 30 . アーリーアダプターの初期の態度分布 $= 10 \sim 30$, 各採用者カテゴリーの態度軌跡の時間発展 (赤 : N_{inn} , 水色 : N_{ea} , 青 : N_{em} , 緑 : N_{lm} , ピンク : N_{lg}). 人的ネットワークはリンク接続確率 0.5 のランダムネットワーク. アーリーアダプター, アーリーマジョリティ, レイトマジョリティ, ラガードからアーリーアダプターへの D_{ij} はランダムに 1 から 2, それ以外の D_{ij} はランダムに -1 から 1 に設定. メディア効果 : 左図 $A(t) = 0$. 右図 $A(t) = 5$.

3.3 まとめ

人間関係に信頼と不信の両方を組み込んだ, Trust-Distrust モデルをベースに, 採用者カテゴリー (イノベーター, アーリーアダプター, アーリーマジョリティ, レイトマジョリティ, ラガード) を組み込んだシミュレーションを実施した. 今回のシミュレーションでは, エージェントを 1000 に設定し, 人々を結ぶ信頼度係数 D_{ij} を 2 パタ

ーンで実行している。1つは D_{ij} を-1~1の乱数に設定した。したがって、 D_{ij} が正の値をとる確率と負の値をとる確率は、それぞれ50%となる。もう一方では、アーリーアダプター、アーリーマジョリティ、レイトマジョリティ、そしてラガードからアーリーアダプターへの信頼係数を1~2の1様乱数に設定した。人から人への接続は、完全なネットワークの一部であるランダムネットワークで、ノード間のリンクの確率は50%に設定した。

各採用者カテゴリーの革新性について採用順序は明確であるが、各採用者カテゴリーの採用曲線については \tanh にて逐次的にシミュレーションに加わるよう設定した。各採用者カテゴリーの詳細な採用曲線については、カテゴリーなどによっても大きく異なることが想定されるので、目安となる軸を設定し、詳細に検討する必要がある。

図21、22ではメディア効果はゼロとし、態度の初期分布の操作を操作したシミュレーションとなっている。図21ではイノベーターを、図22ではアーリーアダプターの初期の態度分布をそれぞれ正と負に設定してシミュレーションを実行している。イノベーターやアーリーアダプターの初期の態度分布が正でも負でも、どちらでスタートさせても、正/負の比率に大きな差をみることが出来なかった。マスメディアの影響や信頼係数を操作しない限り、市場の人々の態度は平準化されていくと推察される。市場がホットなときは油を注ぎ、炎上している場合は(適切な対応は必要だが)過度な対応は取らずとも、いずれ落ち着く可能性を示唆している。

図23はメディア効果をゼロと設定した上で信頼係数 D_{ij} を操作したシミュレーションとなっている。イノベーションの普及において、アーリーアダプターは、オピニオンリーダーの役目を期待されている採用者カテゴリーである。そこでイノベーターを除く採用者カテゴリーはアーリーアダプターを信用する(非負)とした。結果、アーリーアダプターの初期の態度分布が正の場合、正の態度比率が高い傾向にあり、負にある場合は、その逆の傾向にあることを確認した。アーリーアダプターが常に味方であ

る保証はないので、アーリーアダプターを対象としたキャンペーンの実施など、きめ細やかな対応が必要なことを示唆している。

図 24 と 25 では、メディア効果の操作によるシミュレーションとなっている。ここではメディアの効果により、人々の態度はメディアに向かって歪むことが示された。しかしながら今回は、マスメディアやデジタルメディアなど、複数のメディアを想定したシミュレーションまでは実行出来ていないので、その点については今後検討を重ねていく必要がある。

図 25 では、アーリーアダプターの初期の態度分布、信頼係数、そしてメディア効果を加えたシミュレーションとなっている。信頼されているアーリーアダプターが、イノベーションの普及に対してポジティブであり、さらにメディアの影響でプラスに向かってバイアスがかかっていくので、イノベーションの普及に関しては絶好の環境といえる。アーリーアダプターの正の初期態度分布、他採用者カテゴリーからの信頼、正方向へのメディアバイアス。それぞれを構築していくことが重要であると言える。

第4章 普及/伝播のネットワークに関する議論

ここまでの議論では Trust-Distrust モデルを用いて、イノベーションの普及についてシミュレーションを実行している。その際、人々の信頼係数、メディア変数、ランダムネットワークの接続確率が操作されたが、人々のネットワークはランダムネットワークが採用され、その他のネットワークを用いたシミュレーションは実行されなかった。しかしながら実市場における消費者間の繋がりは多種多様であるため、消費者間のつながり方を意識したメディア投下といったマーケティング施策を実行することは重要である。そこで本章では、ランダムネットワークの他に、スケールフリーネットワーク、スモールワールドネットワークを Trust-Distrust モデルに適用しイノベーションの普及についてシミュレーションを試みる。またネットワーク間の違いを比較できるようにするため、リンクの接続確率を調整することで、各ネットワークのリンク数を概ね揃えている。

シミュレーションの状況としては、スケールフリーネットワークは SNS などデジタル空間、例えばアットコスメで多くのフォロワーを抱えるユーザーのレビューによるフォロワーの態度変容、また主に SNS 上で提供されるオンラインゲームでのユーザーのやり取りが上げられる。またスモールワールドネットワークは、フィジカル空間における知人関係、例えば幼稚園や保育園のママ友間での相互作用、スポーツジムでのメンバー同士のインタラクションによる態度変容が想定される。

4.1 シミュレーション1：ベース

まずランダム ネットワーク、スケールフリー ネットワーク、スモール ワールド ネットワークを適用した1 カテゴリー エージェントの Trust-Distrust モデルを使用してシミュレーションを実行する。

4.1.1 ランダムネットワークを使用したシミュレーション

Trust-Distrust モデルにランダム ネットワークを適用したシミュレーション結果を提示している (図 26). このシミュレーションでは, $N = 500$, エージェントのネットワークはリンク接続確率 1%のランダムネットワーク, エージェントのイノベーションに対する態度の初期値は -30 から $+30$ の間で一様乱数により分布させ, エージェントの信頼係数 D_{ij} は -1 から 1 の間の一様乱数, マスメディア効果 $A(t)$ はゼロに設定している. またシミュレーションは 15 ステップとした.

どのノードも同じ確率でリンクするかどうかが決まるため, それぞれのノードが持つリンクの数には極端な差がないというランダムネットワークの特性上, 態度はポジティブとネガティブの両方に分布している. また, マスメディア効果 $A(t)$ が 0 に設定されているため, その点からも偏りがなく正負の値は概ねバランスがとれている.

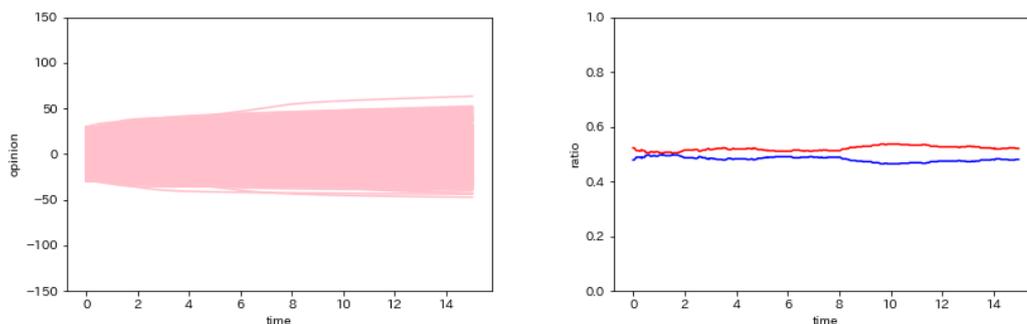


図 26 : $N = 500$ の計算結果. 人的ネットワークはリンク接続確率 0.01 のランダムネットワーク. 左図 : 態度の軌跡の進化を提示. 右図 : 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が負 (-). D_{ij} は -1 から $+1$ までランダムに設定. マスメディア効果はゼロ ($A(t) = 0$).

4.1.2 スケールフリーネットワークを利用したシミュレーション

図 27 では 1 カテゴリーエージェントの Trust-Distrust モデルにスケールフリーネットワークを適用した計算結果を提示している (図 27). シミュレーションに利用したスケールフリーネットワークは, バラバシ・アルバートモデル [35]を $N = 500$,

次数を 1 にて作成した(図 28). 各エージェントの次数の割り当てについてはランダムに設定している. またネットワーク構造以外は, 4.1.1 のランダムネットワークでのシミュレーションと同条件としている. 図 27 右側の折れ線グラフを見ると, 正の線が正, 負の線が負のままで, 構成比を変えずにシミュレーションが完了している. スケールフリーネットワークは多数のエッジを持つ少数のノード (ハブ) と, 少数のエッジを持つ多数のノードで構成されているためハブの影響を受けやすいと考えられるが, エッジ数が少なく意見分布の時間変化が遅くなったためか, 本シミュレーションでは正負の構成比が変化しなかったと考えられる.

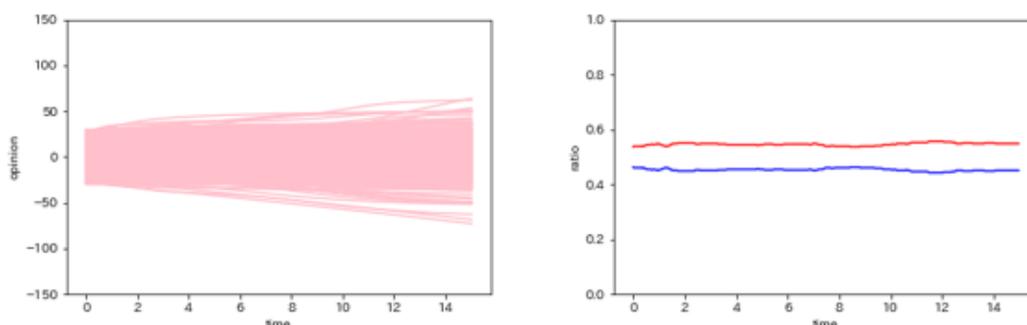


図 27: $N = 500$ の計算結果. ヒューマン ネットワークは, 次数 = 1 のスケールフリー ネットワーク. 左図: 態度の軌跡の進化を提示. 右図: 正 (+) と負 (-) の割合の時系列変化で, 赤が正 (+), 青が負 (-). D_{ij} は -1 から +1 までランダムに設定. メディア効果はゼロ ($A(t) = 0$).

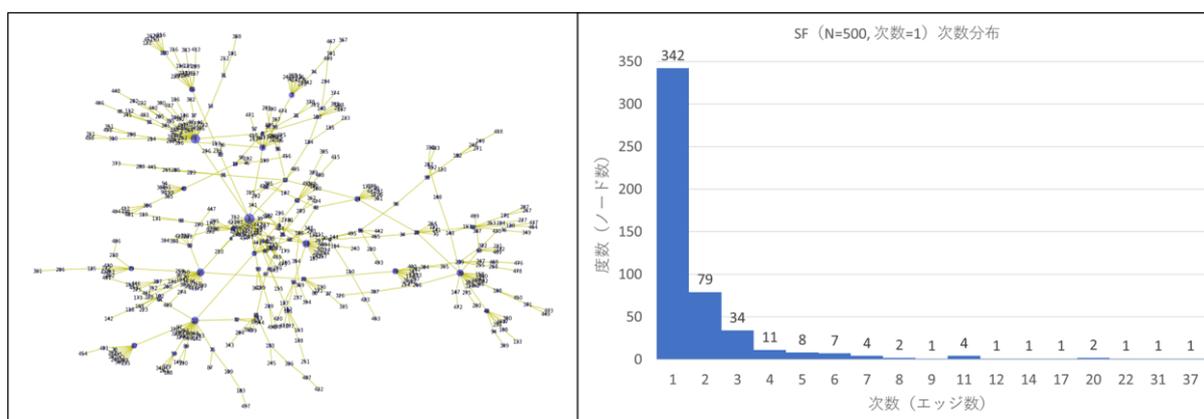


図 28: バラバシ・アルバートモデル ($N = 500$, 次数 1). 左: ネットワーク図. 右: 次数分布

4.1.3 スモールワールドネットワークを利用したシミュレーション

図 29 では 1 カテゴリーエージェントの Trust-Distrust モデルにスモールワールドネットワークを適用した結果を示している(図 29). シミュレーションに使用したスモールワールドネットワークは, ワッツ・ストロガッツモデル [36]を $N = 500$, 次数 = 4, 接続確率 = 10%として作成した(図 30). 各エージェントが持つ次数については, スケールフリーネットワークと同様にランダムで割り付けている. またここでもネットワーク構造を除いて, 設定はランダムネットワークを使用したシミュレーションの場合と同じである. 人々の態度は肯定的な態度と否定的な態度に同程度に分布していることが観察できる.

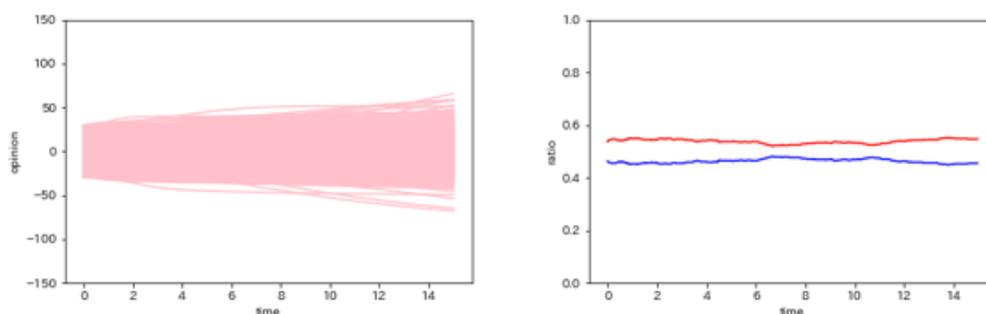


図 29 : $N = 500$ の計算結果. ヒューマン ネットワークは, 次数 $k = 4$, 接続確率 $p = 0.1$ のスモールワールド ネットワーク. 左図: 態度の軌跡の進化を提示. 右図: 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が負 (-). D_{ij} は-1から+1までランダムに設定. メディア効果はゼロ ($A(t) = 0$).

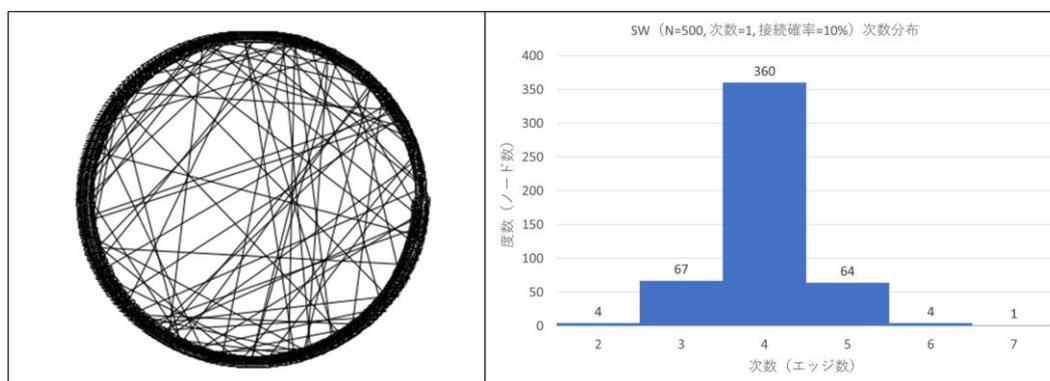


図 30 : ワッツ・ストロガッツモデル ($N = 500$, 次数 4, 接続確率 10%). 左: ネットワーク, 右: 次数分布.

4.1.4 シミュレーション1の比較

表1にランダムネットワーク、スケールフリーネットワーク、そしてスモールワールドネットワークのシミュレーション1を各5回行い、それぞれの態度分布の平均をまとめている。

表1：シミュレーション1のまとめ

	ランダム ネットワーク	スケールフリー ネットワーク	スモールワールド ネットワーク
平均	0.15	0.54	0.77
標準偏差	22.09	20.87	24.20
最大値	62.72	78.17	85.59
最小値	-48.71	-73.93	-82.78
歪度	0.11	-0.12	0.03
尖度	-0.67	0.89	0.66

一見すると3つのシミュレーション結果は、ともに平均が0付近にあり、また標準偏差も20前後と分布する態度は同じように見える。とはいえ3つのシミュレーションの違いを若干ではあるがデータ分布の裾の広がりに見ることができる。態度の幅は最大値と最小値から、スモールワールドネットワーク、スケールフリーネットワーク、ランダムネットワークの順に裾が広がっている。スモールワールドネットワークではスケールフリーネットワークのハブほど中心的な役割を果たすエージェントは存在しないが、近い者同士がつながり態度形成されるので、スケールフリーネットワークと比較してより広く分布する形で複数の態度グループが形成されると考えられる。またスケールフリーネットワークではハブを中心としてエージェント間のインタラクションが起こり態度が形成されるので、ハブごとに比較的まとまった形で態度形成がされると考えられる。ランダムネットワークでは態度形成となる中心的なエージェントが存在せず、エージェント間の相互作用はランダムで発生するので、極端に裾野が

広がることがなかったとったと考えられる。また歪度²と尖度³はともに0に近い値となっているので、左右対称で正規分布に近い分布に近づいていることが伺える。

肯定的な態度と否定的な態度は態度の初期値で決まるようにみえるが、エージェント間のインタラクション、相互作用のしやすさが態度の広がりに影響していると考えられる。

4.2 シミュレーション2：シミュレーション1にメディア変数を加味したシミュレーション

シミュレーション1ではメディア変数をゼロ ($A(t) = 0$) として計算したが、ここではメディア変数を3 ($A(t) = 3$) と設定し、正方向にバイアスをかけたシミュレーションを実行する。またメディア変数以外の変数設定は、シミュレーション1と同じとしている。

4.2.1 ランダムネットワークを使用したシミュレーション

図31では、シミュレーション1のランダムネットワークのシミュレーションにメディア変数 $A(t) = 3$ を追加したシミュレーションを提示している (図31)。メディアによるバイアスは正方向に適用されているため、正の比率はシミュレーションのステップ毎に増加している。シミュレーションを終了させた15ステップでは、エージェントの98%がポジティブ、2%がネガティブとの結果を得た。

² 分布が正規分布からどれだけ歪んでいるかを表す統計量。左に偏った分布のときには正の値、右に偏った分布のときには負の値、そして正規分布の場合は0となる。

³ 分布が正規分布からどれだけ尖っているかを表す統計量。正規分布より尖った分布では正の値、正規分布より扁平な分布では負の値、そして正規分布では0となる。また尖度には2つの定義が存在するが、本論文では0を基準とする尖度を用いている。

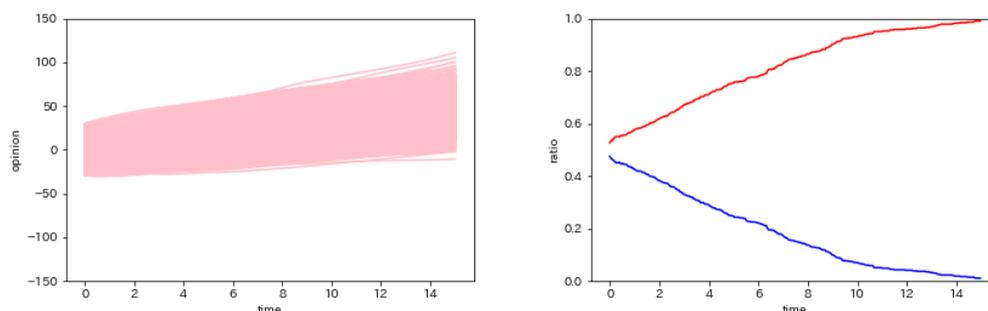


図 31 : $N = 500$ の計算結果. 人的ネットワークはリンク接続確率 0.01 のランダムネットワーク. 左図 : 態度の軌跡の進化. 右図 : 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が負 (-). D_{ij} は -1 から $+1$ までランダムに設定. メディア効果は 3 ($A(t) = 3$).

4.2.2 スケールフリーネットワークを使用したシミュレーション

前項ではシミュレーション 1 のスケールフリーネットワークではメディア効果をゼロ ($A(t) = 0$) としたが, ここでは $A(t) = 3$ としてシミュレーションを実行している (図 32). このシミュレーションも正方向にバイアスが掛かっている為, シミュレーションが進むにつれて正の比率が増加していることがみてとれる. シミュレーションの 5 ステップ目で正の比率が 80% に達していることが確認できた. 大多数が肯定的な態度を持つ集団にあっても, 少数ながらネガティブの態度グループの存在も確認できる. これはハブから距離の遠いエージェントが, ハブの影響を受けていない可能性が考えられる.

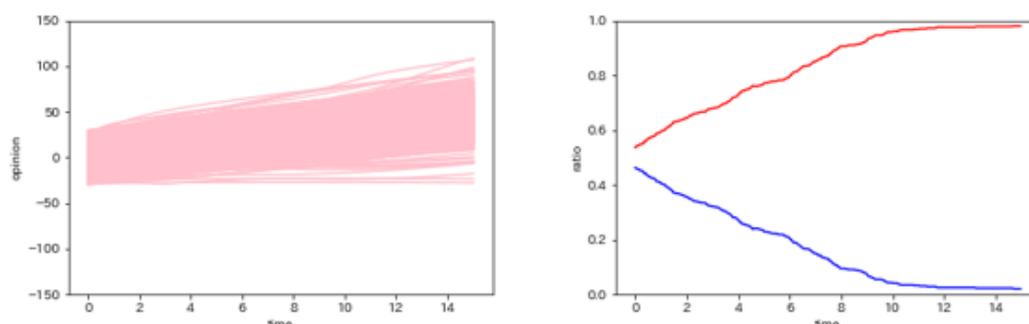


図 32 : $N = 500$ の計算結果. 人的ネットワークは, 平均次数 $= 1$ のスケールフリー ネットワーク. 左図 : 態度の軌跡の進化を提示. 右図 : 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が負 (-). D_{ij} は -1 から $+1$ までランダムに設定. メディア効果は 3 ($A(t) = 3$).

4.2.3 スモールワールドネットワークを使用したシミュレーション

4.2.2と同様に、本項では4.1.3で実行したスモールワールドネットワークを利用したシミュレーションに、メディア効果を3 ($A(t) = 3$)としてシミュレーションを実行している(図33)。このシミュレーションでも正方向にバイアスが掛かっているため、シミュレーションが進むにつれて多くのエージェントの態度はプラス方向に偏っていくことがみてとれる。6ステップまで進むと、正の比率が80%に達することが確認された。最終的な態度分布は、98%がポジティブ、2%がネガティブとなった。

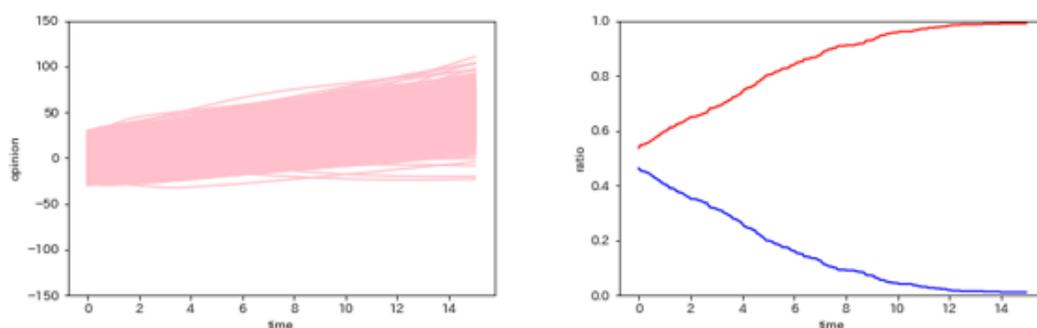


図 33: $N = 500$ の計算結果. ヒューマン ネットワークは, 平均次数 $k = 4$, 接続確率 $p = 0.1$ のスモールワールドネットワーク. 左図: 態度の軌跡の進化を提示. 右図: 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が負 (-). D_{ij} は -1 から $+1$ までランダムに設定. メディア効果は 3 ($A(t) = 3$).

4.2.4 シミュレーション2の比較

表2にランダムネットワーク，スケールフリーネットワーク，そしてスモールワールドネットワークのシミュレーション2を各5回行い，それぞれの態度分布の平均をまとめている。

表2：シミュレーション2のまとめ

	ランダム ネットワーク	スケールフリー ネットワーク	スモールワールド ネットワーク
平均	45.25	45.19	46.40
標準偏差	21.54	20.65	23.68
最大値	100.36	126.74	131.32
最小値	-6.29	-25.93	-21.40
歪度	0.01	0.11	0.29
尖度	-0.70	0.95	0.50

3種類のネットワークのシミュレーション結果は平均が45付近，そして標準偏差が20～23であり，平均と標準偏差に着目するとここでも似たような結果となった。しかしながら最大値と最小値に着目すると，ランダムネットワークでは，他の2つのネットワーク構造とは異なる結果となっている。スケールフリーネットワークとスモールワールドネットワークでは，ランダムネットワークと比較して最大値が25～30大きい値となっている。また最小値でもスケールフリーネットワークとスモールワールドネットワークでは，ランダムネットワークと比較してより小さな値を示している。コミュニケーションの2段階の流れ仮説 [54] [55]が示す通り，ネットワーク内のハブが機能することにより，コミュニケーション効果が高まっているのではないかと推測される。

歪度はいずれのネットワークのシミュレーション結果でも正であったが，スケールフリーネットワークとスモールワールドネットワークの方がより右側のすそが長く左側に山が寄っている分布であるといえる。これは最大値の値からも見ることができる。尖度ではランダムネットワークのみが負の値となった。ランダムネットワークが，

他の二つのネットワークと比較して、平坦な形であることを示していると考えられる。スケールフリーネットワークの方がスモールワールドネットワークと比較して尖度の値が大きいのは、末端のエージェントのエッジ数が少なく、意見分布の時間変化が遅いからと考えられる。

4.3 シミュレーション 3 : 5 アダプターカテゴリーによるシミュレーション

本節ではランダムネットワーク、スケールフリーネットワーク、およびスモールワールドネットワークを使用し、エージェントとして5つのアダプターカテゴリーを採用した Trust-Distrust モデルによるシミュレーションを実行する。

4.3.1 ランダムネットワークを使用したシミュレーション

図 34 が 5 つのアダプターカテゴリーがエージェントに適用され、ネットワークとしてランダムネットワークが採用された Trust-Distrust モデルのシミュレーション結果である。エージェント数は 1000 であり、5 つのアダプターカテゴリーのエージェントの構成比は Rogers に倣い、イノベーター : $n = 25$ (2.5%), アーリーアダプター : $n = 135$ (13.5%), アーリーマジョリティ : $n = 340$ (34.0%), レイトマジョリティ : $n = 340$ (34.0%), ラガード : $n = 160$ (16.0%)としている [14]。また図 18 で示されているように、各エージェントは時差を持ってシミュレーションに投入されている。最初にシミュレーションに投入されるイノベーターの態度の初期値は ± 30 の範囲でランダムに分布するよう設定し、他のエージェントの初期値はゼロに設定している。エージェント間の相互信頼係数 D_{ij} は、 -1 から $+1$ の間の一様乱数によって決定される。

今回のシミュレーションでは、マスメディア効果はゼロに設定しているため、正または負方向へのバイアスは掛かっていない。したがってシミュレーション終了時の肯定的な態度と否定的な態度の比率は非常に近いものになっている。採用曲線は S 字型

であり、これはエージェントを1カテゴリーのみに設定したシミュレーションでは観察できなかった。この結果は本モデルがロジャースの仮説 [14]通りになる設定になっていることを示した位置づけであり、エージェントの5つのアダプターカテゴリーがシミュレーションに順次投下されること、つまり各エージェントのイノベーションに対する採用速度が異なることによって引き起こされているということである。

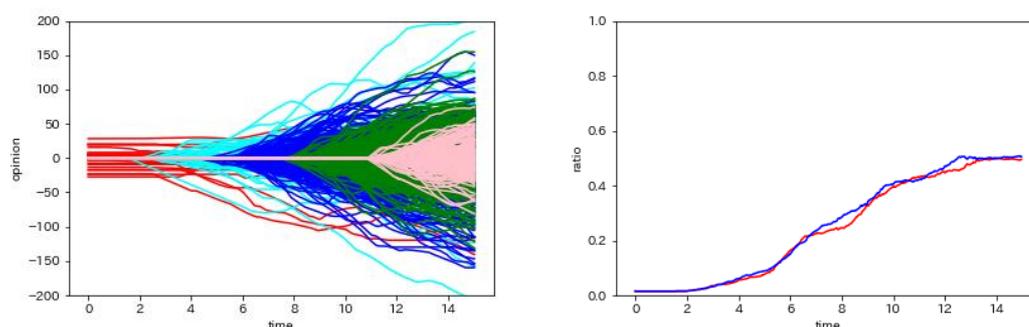


図 34: $N = 1000$ の計算結果. 人のネットワークはリンク接続確率 0.01 のランダム ネットワーク. 左図: 各採用者カテゴリー(赤: N_{inn} , 水色: N_{ea} , 青: N_{em} , 緑: N_{lm} , ピンク: N_{lg})の態度の軌跡の時間発展を提示. 右図: 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が負 (-). D_{ij} は -1 から $+1$ までランダムに設定. メディア効果はゼロ($A(t) = 0$).

4.3.2 スケールフリーネットワークを利用したシミュレーション

図 35 はスケールフリーネットワークを Trust-Distrust モデルのネットワークとして設定したシミュレーション結果となっている(図 35). シミュレーションに利用したスケールフリーネットワークは、バラバシ・アルバートモデルを $N = 1000$, 平均次数 1 にて作成した(図 36). パラメータの設定はネットワーク構造を除いて、ランダムネットワークを使用したシミュレーション 4.3.1 と同じである。

Rogers [14]によると、アーリーアダプターはオピニオンリーダーとして機能することが期待されているため、多くの次数を持つハブ(上位 135)をアーリー アダプターとして設定し、またアーリーアダプター以外のエージェントの次数についてはランダムで設定している。しかし、このシミュレーションではエージェント間の信頼係数

もコミュニケーション効果変数を含めて中立に設定されているため、肯定的な態度と否定的な態度の比率が近くなっている。エージェントの肯定的な態度と否定的な態度は凡そ同数となっているが、エージェントの態度の分布は広がっていないようにみえる。これは他のシミュレーション結果には見られなかった特徴といえる。次数が低くハブやその他のエージェントからの影響を受けにくいエージェントがゼロ付近に存在しており、これらエージェントがラガードの可能性が高い。

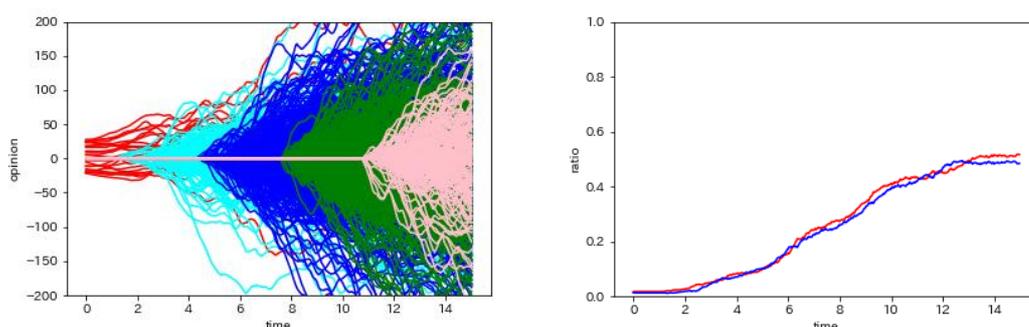


図 35 : $N = 1000$ の計算結果。ネットワークは、平均次数 1 のスケールフリーネットワーク。左図：各アダプター カテゴリーの態度の軌跡の時間発展を提示(赤 : N_{inn} , 水色 : N_{ea} , 青 : N_{em} , 緑 : N_{lm} , ピンク : N_{lg})。右図：正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が負 (-)。 D_{ij} は -1 から +1 までランダムに設定。メディア効果はゼロ ($A(t) = 0$)。

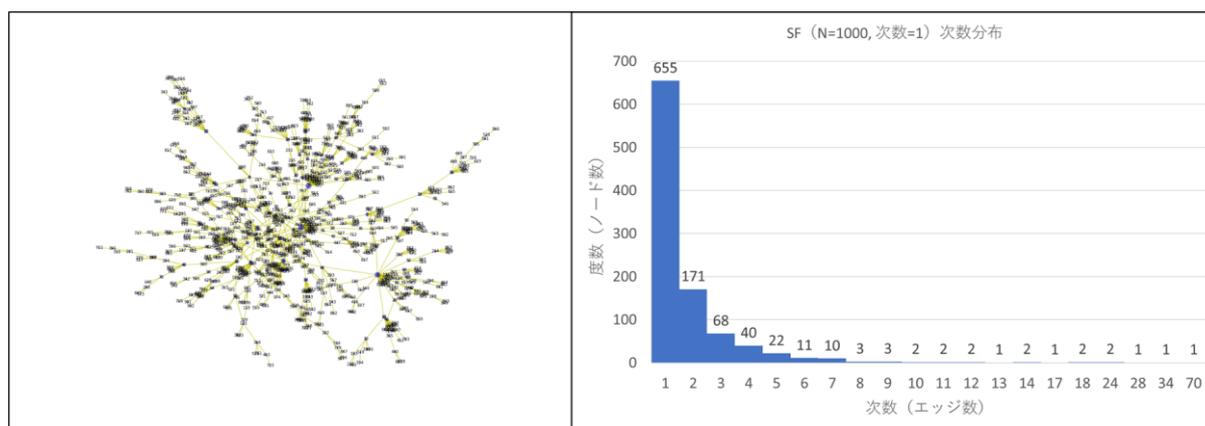


図 36 : バラバシ・アルバートモデル ($N = 1000$, 平均次数 1)。左 : ネットワーク。右 : 次数分布

4.3.3 スモールワールドネットワークを使用したシミュレーション

図 37 は、スモールワールドネットワークを Trust-Distrust モデルのネットワークに設定したシミュレーション結果となっている(図 37)。シミュレーションに使用したスモールワールドネットワークは、ワッツ・ストロガッツモデルを $N = 1000$, 次数 $= 4$, 接続確率 10%として作成した(図 38)。各エージェントが持つ次数については、スケールフリーネットワークと同様にリンク数の多い上位 135 のハブ (エージェント) をアーリーアダプターとして設定している。またここでもネットワーク構造を除いて、設定はランダムネットワークを使用したシミュレーションの場合と同じである。

このシミュレーションでもエージェントの意思決定にバイアスを与えるメディア変数の設定をしていないため、エージェントの態度は肯定的態度と否定的態度の間でほぼ等しくなっている。

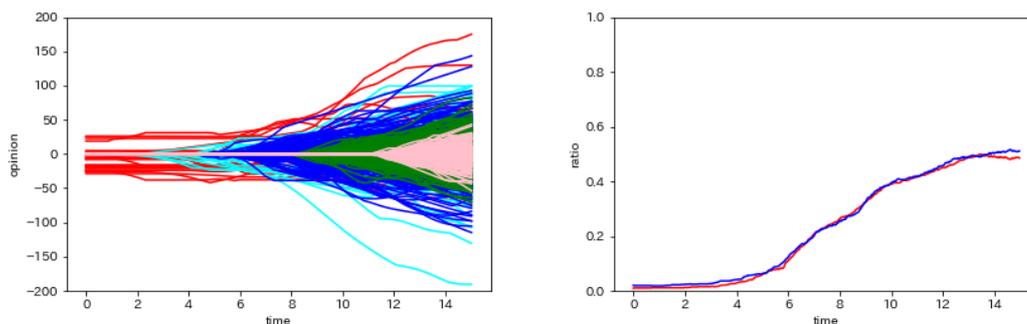


図 37 : $N = 1000$ の計算結果. ヒューマン ネットワークは平均次数 $k = 4$, 接続確率 $p = 0.1$ のスモールワールド ネットワーク. 左図 : 各採用者カテゴリー(赤 : N_{inn} , 水色 : N_{ea} , 青 : N_{em} , 緑 : N_{lm} , ピンク : N_{lg})の態度の軌跡の時間発展を提示. 右図 : 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+) , 青が負 (-) . D_{ij} は-1から+1までランダムに設定. メディア効果はゼロ($A(t) = 0$).

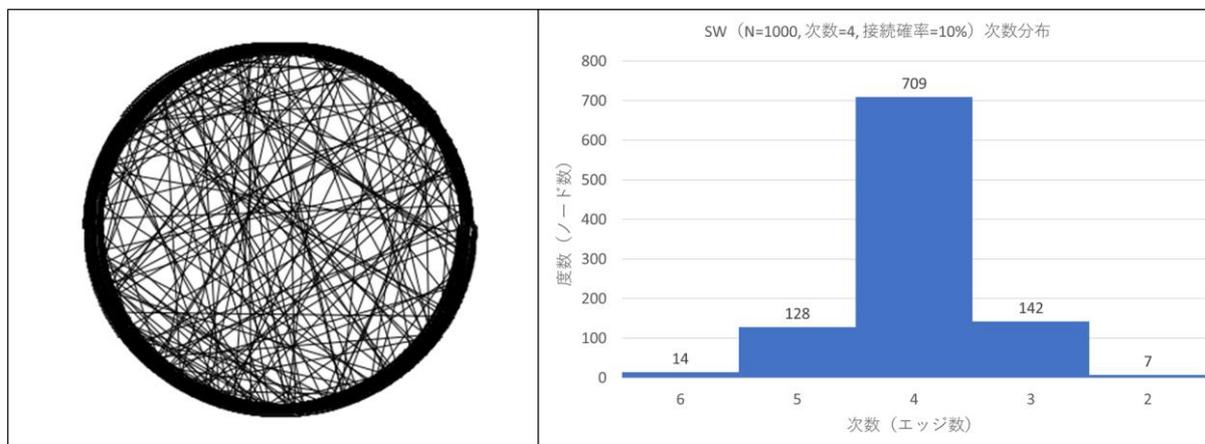


図 38 : ワッツ・ストロガッツモデル ($N = 1000$, 平均次数 4, 接続確率 10%). 左 : ネットワーク. 右 : 次数分布

4.3.4 シミュレーション 3 の比較

表 3 にランダムネットワーク, スケールフリーネットワーク, そしてスモールワールドネットワークの 5 アダプターカテゴリーによるシミュレーションを各 5 回行い, それぞれの態度分布の平均をまとめている.

表 3 : シミュレーション 3 のまとめ

	ランダムネットワーク	スケールフリーネットワーク	スモールワールドネットワーク
平均	-1.09	-0.30	0.22
標準偏差	43.42	94.74	30.63
最大値	171.84	384.18	139.61
最小値	-187.83	-417.46	-165.44
歪度	-0.13	-0.04	-0.03
尖度	1.72	1.49	2.60

最終的な態度分布に, ネットワーク構造を変えた 3 つのシミュレーションの主な違いをみることが出来る. スケールフリーネットワークを使用したシミュレーションでの態度の分布は, より分散しているが, スモールワールドネットワークを用いたシミュレーションでは, 標準偏差の値が小さくデータが散らばっていない.

態度分布の違いは、他者により影響を与えるハブからの距離、各エージェント間の距離がネットワーク毎に異なるため、それがエージェントの態度形成に影響を与えていると考えられる。また5アダプターカテゴリーによるシミュレーションの場合、時差をもってエージェントがシミュレーションに参加するため、遅れてシミュレーションに参加する革新性の低いエージェント（レイトマジョリティやラガード）は、やはり態度変容を起こしにくいと考えられる。これは広告キャンペーンは期間を決めて展開されるので、その期間のうちに市場の隅々にまで態度変容を促すよう、メッセージを届けることが難しいことを示している。

また歪度は3つのタイプのネットワークともに0に近い値となっているため、左右対称の分布であることが伺える。メディア効果はゼロなので、これについては妥当な結果と言える。尖度については3つのネットワークとも正なので、尖った形であることが伺える。これはシミュレーションに後から参加してくるエージェントの意見分布の時間変化が遅いことが考えられる。実社会における広告キャンペーンも期間を区切って展開されることから、革新性や情報感度の低い消費者への働きかけは必須といえる。

4.4 まとめ

今回のシミュレーションでは、ネットワーク構造が違っていても、態度の平均値や標準偏差、歪度、尖度などから、同じような時間的发展をすることが伺いしれた。ネットワーク構造が違っていても、他の消費者そしてメディアに影響され、消費者は自身の態度を形成していくことが確認できた。

その一方で最大値や最小値など態度の広がり幅については、ハブの存在や他エージェントとの距離などネットワーク構造の違いが態度分布の違いを生んでいることも示唆された。

またエージェントの態度は、メディア（外生）変数に加え、対面での情報交換など他者との接触によって影響を受けると考えられるが、遅れてシミュレーションに参加するエージェントには他エージェントと接触する時間が短いので、先行してシミュレーションに参加しているエージェントと比較してより0付近に留まっている傾向にある。その点が1カテゴリーと5アダプターカテゴリーによるシミュレーションの差に表れていると考えられる。マーケティング・コミュニケーションの文脈では、限られたマーケティング・キャンペーン期間中にいかに早く広告といった情報に触れ、対象となる商品やサービスに好意的な態度をもつ他の消費者と相互作用を起こさせるかがカギとなるであろう。

第5章 メディア効果に関する議論

SNS の普及以前はマスメディアから発信される情報が社会の広範囲に影響を与え、人々のクチコミはリアルに接触が可能な狭い範囲に限られていた。しかし SNS の普及によりクチコミの影響範囲は、マスメディアからもたらされる情報の範囲となんら遜色のない範囲にまで広がっている [39] [56]。かねてよりイノベーションの普及や人々の態度形成に影響を与えるのはマスコミからの情報とクチコミであるとされているから、クチコミ（人々のインタラクション）は、その重要性を増していると考えられる。

新しい情報やイノベーションはメディアによってもたらされ、人々へと拡散していく。つまりメディアからもたらされる情報と個々人のインタラクションの双方が、イノベーションの普及に影響を及ぼしている。情報を発信する側としては、狙ったターゲットにメッセージを発信し、その人たちからポジティブなクチコミを他者に拡散してもらえるのが理想といえる。どのようにメッセージを発信すればよいのかを知ることとは重要な点であろう。

本章では Trust-Distrust モデルに異なるリーチパターンを用いたシミュレーションを実行し、態度分布や採用意向の形成過程の違いについて考察する。

5.1 リーチパターンについて

広告コミュニケーションは、消費者の態度変容を促すために、ターゲットとする消費者に広告が複数回到達するよう設計される。リーチパターンとは、広告キャンペーンの期間中に、有効フリークエンシーの範囲内で到達するターゲットを最大化させるための広告継続パターンである。ちなみにリーチとは当該広告を見た人数であり、フリークエンシーは同一のターゲットが当該広告に接触した回数である。さらに有効フリークエンシーは、期待する広告効果を得るのに適した広告接触回数である [54]。

リーチパターンには8つの基本的な型がある。そのうち4種(電撃型, くさび型, 逆くさび・対人影響型, 短期流行型)が新製品向けで, 残りの4つ(インパクト・スケジュール型, ブランド認知維持型, ターゲット移行型, 季節需要刺激型)が既存製品に向けたパターンとされている。本稿では新製品に適しているとされている4つのリーチパターンでシミュレーションを実行する [54] [57]。

以下が新製品に適しているとされている4つのリーチパターンの概要となる。またあくまで相対的なものであるが, ターゲットに対するリーチが棒グラフの棒の高さで示され, 広告キャンペーン中のフリークエンシーは棒グラフの幅で表現されている。

5.1.1 電撃型

電撃型は, 最小有効フリークエンシーに達する連続的広告を広告プロモーション期間中に投下する。リーチとフリークエンシーともに高水準で広告キャンペーンが継続されるリーチパターンである(図 39)。

これは新しいカテゴリーで先発ブランドの優位を獲得したり, 後発ブランドが市場リーダーに対抗したりするような場合に適したリーチパターンであるとされている。

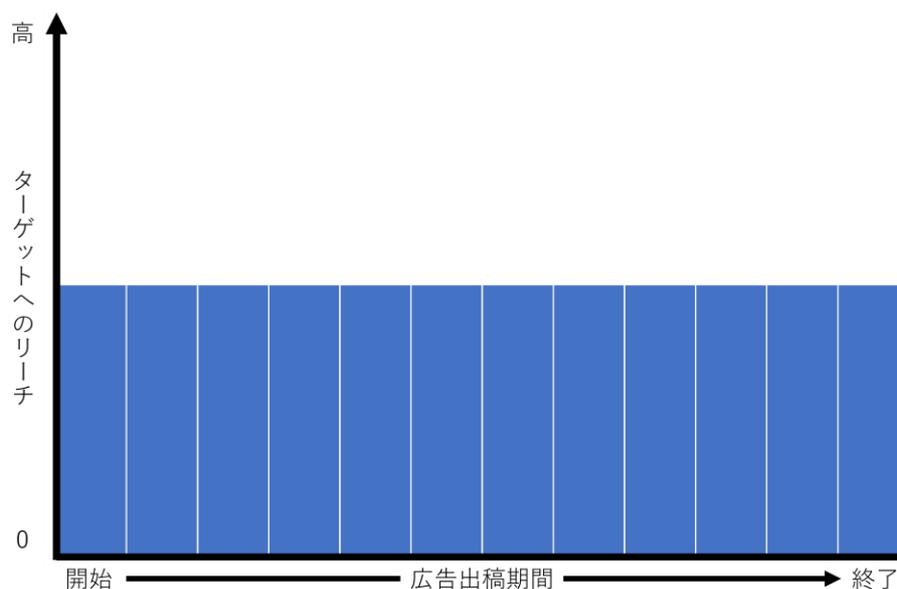


図 39 : 電撃型

5.1.2 くさび型

くさび型は、次第に広告支出を減少させる新製品に典型的なパターンである。リーチは一定でフリークエンシーを徐々に減少させることにより広告量を減少させる(図 40)。これは繰り返し購買される製品やカテゴリーの新ブランド導入に、非常に優れたリーチパターンである [54] [57]。

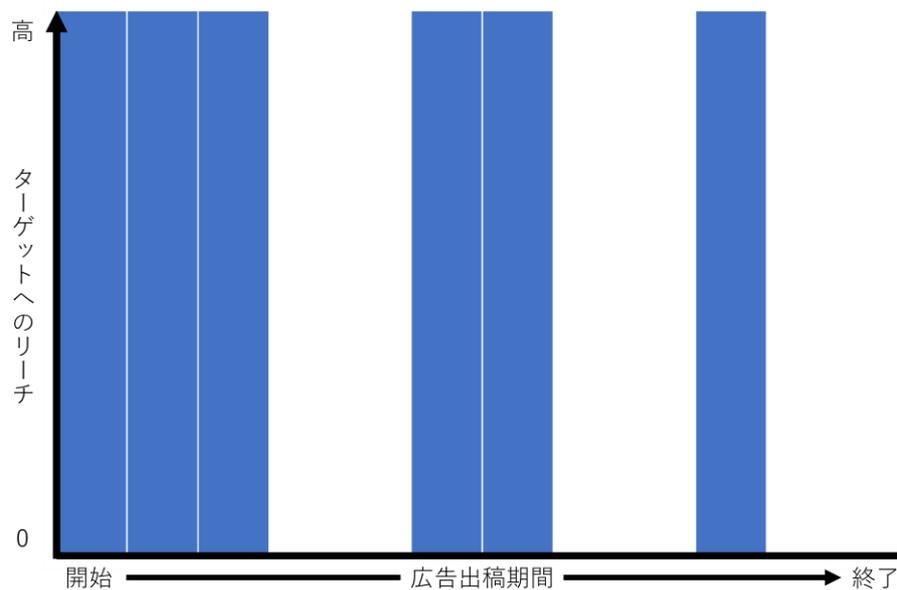


図 40：くさび型

5.1.3 逆くさび・対人影響型

逆くさび・対人影響型は、くさび型とは逆にフリークエンシーを増加させるリーチパターンである。このパターンは対人的影響を経て普及する製品に向いているとされる(図 41)。つまり広告キャンペーンの初期段階では、オピニオンリーダーと目されるターゲットに向けて広告を発信し、彼ら/彼女らから他の消費者に対する情報発信や影響力に期待するパターンである。

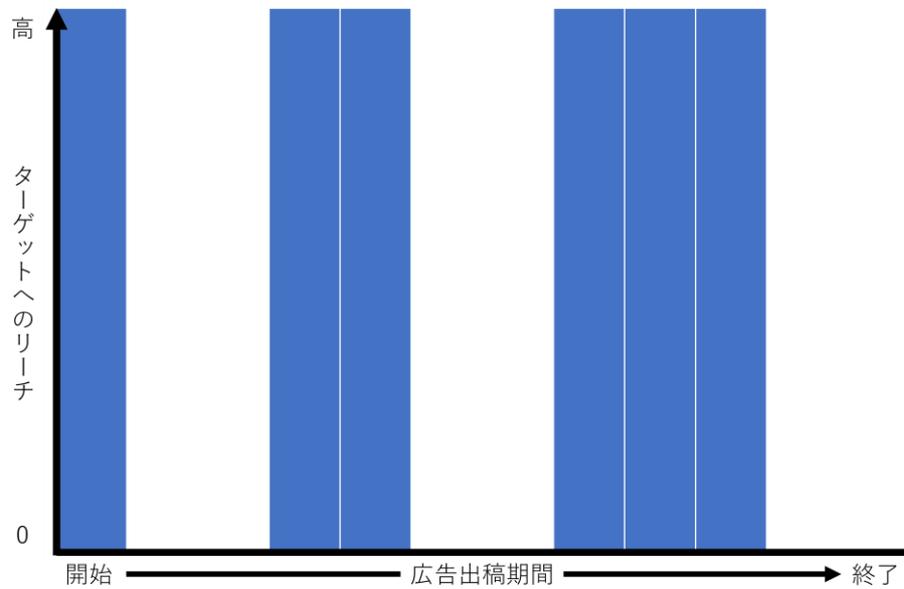


図 41：逆くさび・対人影響型

5.1.4 短期流行型

短期流行型は製品ライフサイクルの短い流行製品向けのリーチパターンである(図42)。ファストファッション、新しい玩具、新しい映画などが流行製品にあたる。このパターンは短期の電撃型に似ているが、流行を短期に広げるため、幅広いリーチと高いフリークエンシーを必要としている。

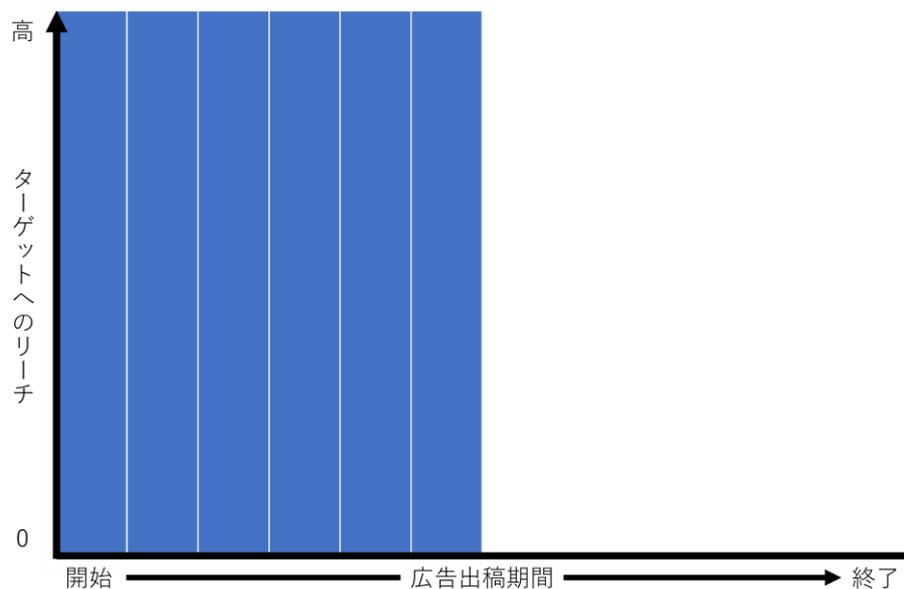


図 42：短期流行型

5.2 シミュレーションの実行

Trust-Distrust モデルにおけるメディア変数 $A(t)$ は、広告の他、パブリシティやクチコミ、季節性など人々の態度形成に影響を与える外部要因全てを指している。全ての外部からの要因を変数としてシミュレーションに組み込むことは困難なので、本研究では5.1で記述した通り電撃型、くさび型、逆くさび・対人影響型、短期流行型を用いてシミュレーションを実行する。

また本稿ではリーチパターンの違いによる態度の収束過程の違いを確認するのでリーチは100%、そして投下される広告量は全て同量とする。

5.2.1 1 カテゴリーエージェントを採用したシミュレーション

本章ではまず1カテゴリーのエージェントによるシミュレーションを、前節で記述した4つのリーチパターンで広告を投下させたパターンに加え、メディア変数無しのバリエーションで実行する。

Trust-Distrust モデルの設定としては、 $N = 300$ 、初期態度は $-30 < I(t) \leq 30$ から乱数で設定、エージェント間の相互信頼係数 D_{ij} は、 -1 から $+1$ の間の一様乱数によって決定され、ランダムネットワークの接続確率は15%である。またシミュレーションのステップ数は12としている。本章でのシミュレーション数を12ステップとした理由としては、日本国内のCMまたは番組の放送契約期間は通常3ヶ月（1クール⁴）とされる点、また1年を月次でみた場合など広告効果分析として適合と考えたからである。

5.2.1.1 マスメディア変数ゼロのシミュレーション

メディア変数を投下していないシミュレーションの結果が図43である。図43からわかるように、人々の態度はポジティブとネガティブであるが、人々の態度は完全

⁴ 1クールは一般的には3ヶ月、13週であるが、月次視点も加味し分析ステップは12とした。

に散らばっている。この計算では、メディア効果 $A(t)$ を0と設定しているため、この図の態度分布はバイアスがなく正と負になる。300人を想定した計算では、エージェントの態度は肯定的な態度と否定的な態度の間ではほぼ等しくなっている。

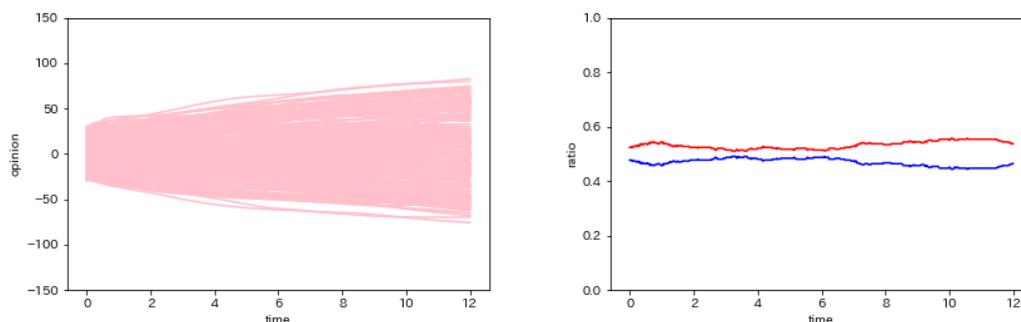


図 43: $N = 300$ の計算結果。人的ネットワークはリンク接続確率 0.15 のランダムネットワーク。左図：態度の軌跡の時間発展。右図：正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が負 (-)。 D_{ij} は-1から+1までランダムに設定。メディア効果はゼロ($A(t) = 0$)。

5.2.1.2 電撃型によるシミュレーション

ここではメディア変数を電撃型にて投入したシミュレーションを実行している(図 44)。図からわかるように、人々の態度はポジティブ方向に分布していることが確認できる。この計算では、マスメディア効果 $A(t) = 5$ をシミュレーションのステップごとに投下している。シミュレーションは12ステップで実行されているので、トータルの広告量は60となっている。シミュレーションでは正方向にバイアスを掛けており、ステップごとにマスメディア変数が投下されているので、リニアに正の比率は高まり、負の比率は低下していることが見て取れる。認知を高めるため、継続的に広告を発信する必要があるケースに向いてる。

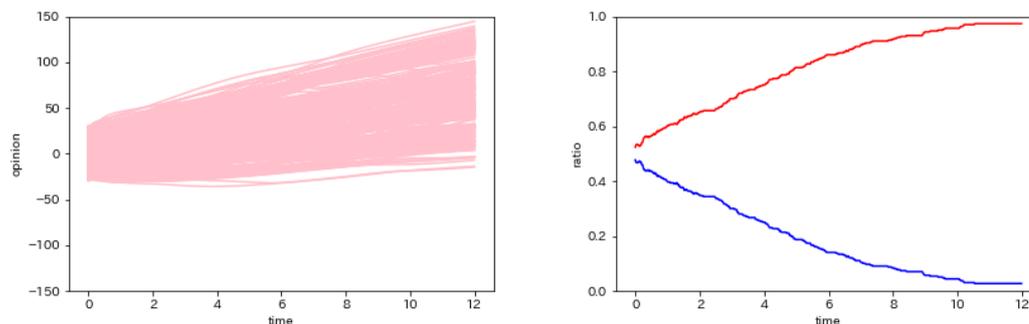


図 44 : $N = 300$ の計算結果. 人的ネットワークはリンク接続確率 0.15 のランダムネットワーク. 左図 : 態度の軌跡の時間発展. 右図 : 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が負 (-). D_{ij} は -1 から +1 までランダムに設定. メディア効果は 5 ($A(t) = 5$). 広告の投下パターンは電撃型.

5.2.1.3 くさび型によるシミュレーション

ここではメディア変数をくさび型にて投入したシミュレーションを実行している (図 45). 本シミュレーションでは, マスメディア効果 $A(t) = 10$ が, ステップ 1, 2, 3, 6, 7, そして 10 の時に投下されている. マスメディア効果 $A(t) = 10$ が計 6 回投下されているので, トータルの広告量は 60 となっている. メディア変数が投下されたタイミング, シミュレーションのステップ 10 では分かりにくい, ステップ 1 から 3, 6 と 7 で正の比率が高まっていることが見て取れる. とくに最初の 3 ステップで連続的に広告が投下された結果, 一気に正方向の態度の人々を獲得できていることが確認できる. メディア変数の投下されないタイミングでマイナスの比率も僅かであるが上昇していることも確認できるが, シミュレーションが進むにつれ, その傾向が弱くなってきている. 広告キャンペーンの初期のタイミングで大量に出稿し, 次第に広告量を減少させる, 新製品に典型的なパターンといえる.

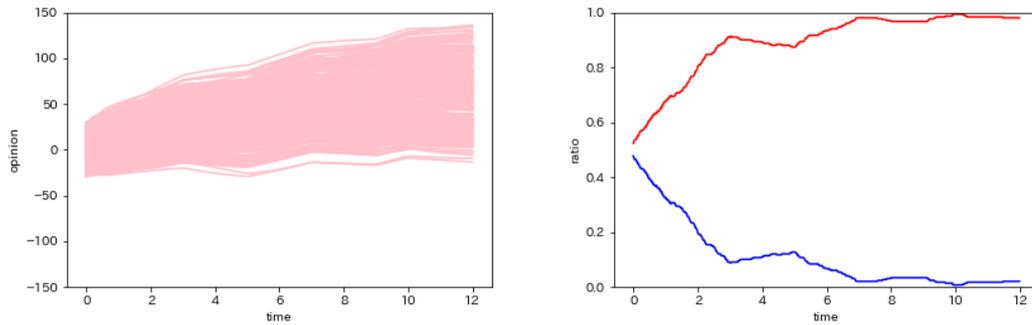


図 45 : $N = 300$ の計算結果. 人的ネットワークはリンク接続確率 0.15 のランダムネットワーク. 左図 : 態度の軌跡の時間発展. 右図 : 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が負 (-). D_{ij} は -1 から +1 までランダムに設定. メディア効果 $A(1, 2, 3, 6, 7, 10) = 10$. $A(4, 5, 8, 9, 11, 12) = 0$. 広告の投下パターンはくさび型.

5.2.1.4 逆くさび・対人影響型によるシミュレーション

ここではメディア変数を逆くさび・対人影響型にて投入したシミュレーションを実行している (図 46). マスメディア効果 $A(t) = 10$ は, ステップ 1, 4, 5, 8, 9, 10 の時に投下されている. くさび型時のシミュレーションと同様に, マスメディア効果 $A(t) = 10$ が計 6 回投下されているので, トータルの広告量は 60 となっている. ステップ 1, 4 と 5, 8-10 と, 広告が投下されたタイミングで徐々に正の比率が高まっていることが見て取れる. 逆に広告が投下されていないタイミングでは, エージェント間の相互作用のみで人々の態度が決まるため, マイナスの態度のエージェントが多い時には, マイナスの態度を持つエージェントも増えるが, プラスの態度を持つエージェントが大勢を占めるようになるとその傾向が弱くなってきていることが確認できる.

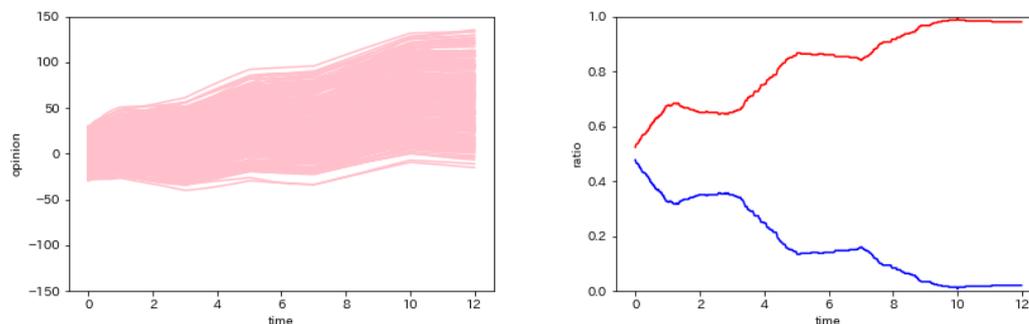


図 46 : $N = 300$ の計算結果. 人的ネットワークはリンク接続確率 0.15 のランダムネットワーク. 左図 : 態度の軌跡の時間発展. 右図 : 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が負 (-). D_{ij} は -1 から +1 までランダムに設定. メディア効果はシミュレーション・ステップ 1, 4, 5, 8, 9, 10 の時 10, ステップ 2, 3, 6, 7, 11, 12 の時 0. $A(1, 4, 5, 8, 9, 10) = 10$. $A(2, 3, 6, 7, 11, 12) = 0$. 広告の投下パターンは逆くさび・対人影響型.

5.2.1.5 短期流行型によるシミュレーション

ここではメディア変数を短期流行型にて投入したシミュレーションを実行している (図 47). マスメディア効果 $A(t) = 10$ がステップ 1 から 6 までの間に投下されているので, トータルの広告量はここでも 60 となっている. 他のリーチパターンとは違い, 大量の広告が短期間に投下されているので, ポジティブの態度を持つエージェントの比率はステップ 5 の段階で既に 96% を超えている. プロダクトライフサイクルの短い流行商品を一気に認知させるのに適しているといえる.

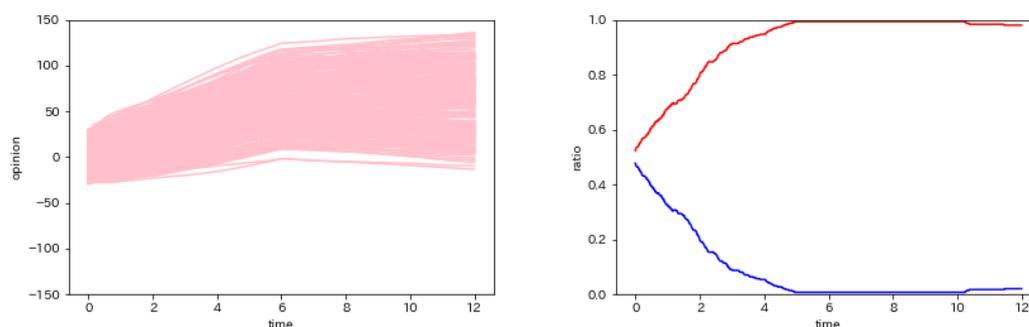


図 47 : $N = 300$ の計算結果. 人的ネットワークはリンク接続確率 0.15 のランダムネットワーク. 左図 : 態度の軌跡の時間発展. 右図 : 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が負 (-). D_{ij} は -1 から +1 までランダムに設定. メディア効果 : $A(1-6) = 10$. $A(7-12) = 0$. 広告の投下パターンは短期流行型.

5.2.1.6 1 カテゴリーエージェントのシミュレーション間の比較

ここでは、各シミュレーションのポジティブの比率の推移を比較する(図 48)。

約 50%からスタートしたポジティブとネガティブの構成比は、12 ステップ目でのプラスの構成比が全 5 リーチパターンとも 97%であった。しかしながらポジティブとネガティブの構成比の推移は、各リーチパターンの特徴を表していると考えられる。

電撃型はシミュレーションのステップごとに広告が投下されているので、リニアに上昇していることが確認できる。一方で一気に構成比を高めたのは短期流行型で、広告を打ち終えたステップ 6 のタイミングでその構成比が 99%となっている。しかしステップ 7 以降で僅かではあるが構成比が下がりだしているのも事実である。

くさび型は、ステップ 3 まで短期流行型と同じ動きをしているが、ステップ 4 と 5 でマスメディア変数がなくなるとプラスの構成比は下がり、マスメディア変数が投入されるステップ 6 と 7 でまた上がっている様子が確認できる。逆くさび・対人影響型はくさび型と逆の動きをしていることが確認できる。

リーチパターンごとにポジティブとネガティブの構成比の推移が異なるので、商品特性と広告予算に応じて、広告キャンペーンの期間中にタイミングを見計らいながら出稿計画を立てることが重要であるといえる。

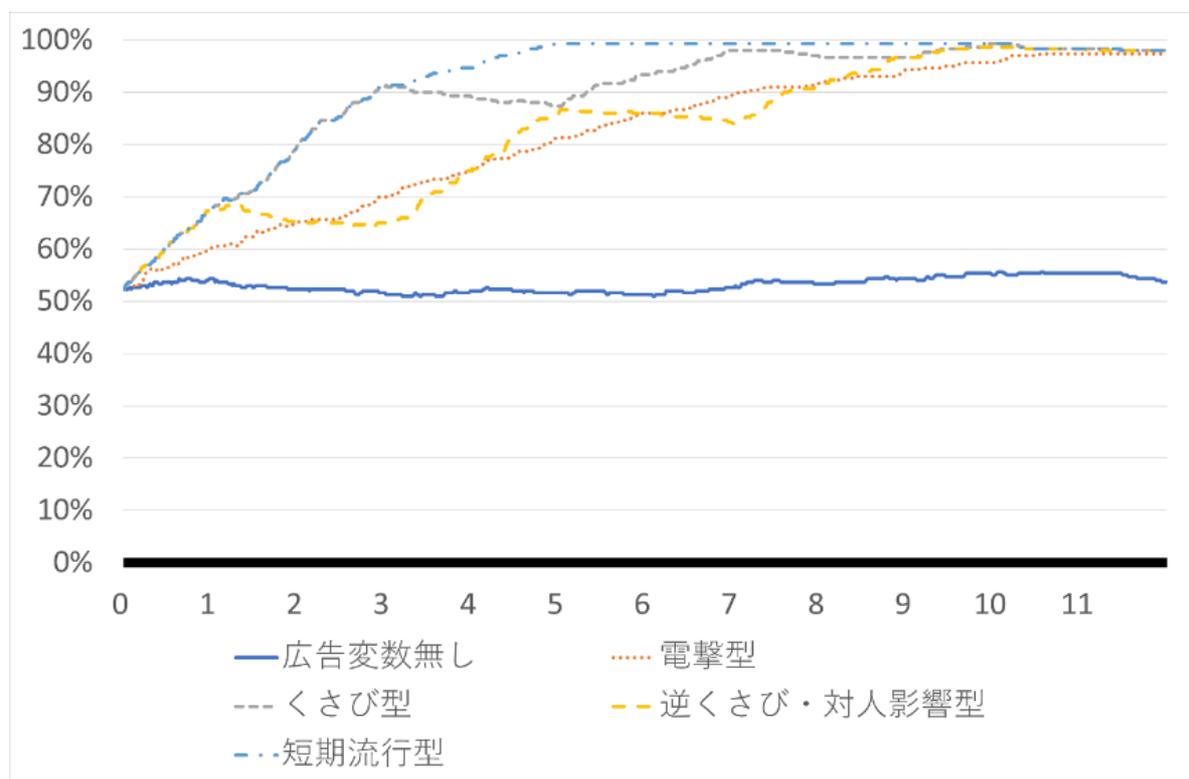


図 48 : 1 エージェントの Trust-Distrust モデルを用いたリーチパターン別ポジティブ構成比の推移

5.2.2 5 カテゴリーエージェントを採用したシミュレーション

本項では5 カテゴリーエージェントによるシミュレーションを, 先に記述した4つのリーチパターンに加え, メディア変数無しのバリエーションでも実行する。

Trust-Distrust モデルの設定としては, $N = 1000$, イノベーターの初期態度は -30 から $+30$ の間の一様乱数によって決定し, アーリーアダプターの態度の初期値は 10 から 20 の間の一様乱数によって決定する. アーリーマジョリティとレイトマジョリティからアーリーアダプターへの信頼係数 D_{ij} は 1 から 2 の間の一様乱数によって決定し, それ以外の D_{ij} は -1 から $+1$ の間の一様乱数によって決定する. この設定によって, 市場からオピニオンリーダーと目されるアーリーアダプターが, 当該イノベーションに正の態度を持ち, また各マジョリティから信頼されているという, イノベーションに対して好意的な市場環境下でのシミュレーションが実行できる. ランダムネットワークの接続確率は 15% である. シミュレーションのステップ数はここでも 12 である.

5.2.2.1 マスメディア変数ゼロのシミュレーション

図 49 はメディア変数を 0 にてシミュレーションを実行した結果である (図 49)。
図 49 からわかるように, 人々の態度はある程度広がり均衡しているが, 最終的な態度の分布は若干プラス方向によっている. これはシミュレーションの前提条件が理由としてあげられる. メディアによるバイアスはないものの, アーリーアダプターの初期態度が 10 から 20 であることと, アーリーマジョリティとレイトマジョリティがアーリーアダプターを信用していることが理由としてあげられる. つまりオピニオンリーターと目されているアーリーアダプターが市場から信頼されていて, そのアーリーアダプターが当該イノベーションに正の態度を有しているので, プラス方向に振れていると解釈できる.

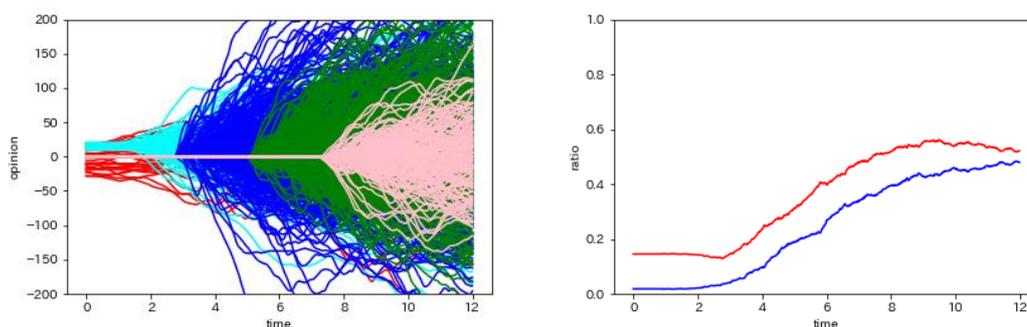


図 49: $N = 1000$ の計算結果. 人的ネットワークはリンク接続確率 0.15 のランダムネットワーク. イノベーターの初期態度は $-30 \leq I(t) \leq 30$ で一様乱数で設定. アーリーアダプターの態度初期値は $10 \leq I(t) \leq 20$ で一様乱数で設定. その他のエージェントの態度の初期値は 0. 左図: 各採用者カテゴリー(赤: N_{inn} , 水色: N_{ea} , 青: N_{em} , 緑: N_{lm} , ピンク: N_{lg})の態度の軌跡の時間発展を提示. 右図: 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が不正 (-). アーリーマジョリティとレイトマジョリティからアーリーアダプターへの信頼係数 D_{ij} は 1 から 2 の間で一様乱数によって設定, それ以外のエージェント間の信頼係数 D_{ij} は -1 から 1 の間で一様乱数によって決定. メディア変数はゼロ ($A(t) = 0$).

5.2.2.2 電撃型によるシミュレーション

ここでのシミュレーションではマスメディア効果 $A(t) = 5$ がステップ1から12までステップごとに投入されている。このことから正の比率は高まり、負の比率は低下していることが見て取れる。ただしその傾向はアーリーマジョリティがシミュレーションに参加するタイミングからである。それ以前ではマイナスの態度を持ったイノベーターとの相互作用が原因で、メディア変数に対してリニアな反応が得られてないと考えられる(図50)。

広告キャンペーン通期で広告を投下し続けるには大きなコストが掛かるが、おおよそ80%のエージェントが正の態度を形成している為、その効果は大きいと考えられる。

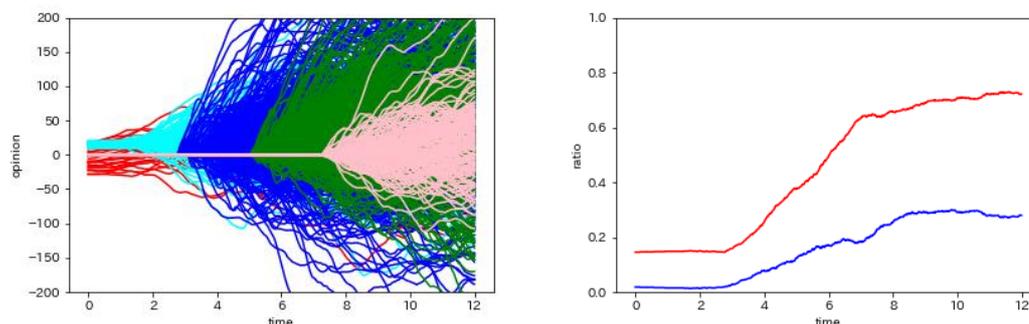


図 50 : $N = 1000$ の計算結果。人的ネットワークはリンク接続確率 0.15 のランダムネットワーク。イノベーターの初期態度は $-30 \leq I(t) \leq 30$ で一様乱数で設定。アーリーアダプターの態度初期値は $10 \leq I(t) \leq 20$ で一様乱数で設定。その他のエージェントの態度の初期値は 0。左図：各採用者カテゴリー(赤： N_{inn} 、水色： N_{ea} 、青： N_{em} 、緑： N_{lm} 、ピンク： N_{lg})の態度の軌跡の時間発展を提示。右図：正 (+) と負 (-) の割合の時系列変化、赤が正 (+) , 青が不正 (-) 。アーリーマジョリティとレイトマジョリティからアーリーアダプターへの信頼係数 D_{ij} は 1 から 2 の間で一様乱数によって設定、それ以外のエージェント間の信頼係数 D_{ij} は -1 から 1 の間で一様乱数によって決定。マスメディア効果 $A(t) = 5$ 。広告の投下パターンは電撃型。

5.2.2.3 くさび型によるシミュレーション

ここではメディア効 $A(t) = 10$ が、ステップ1から3, 6と7, そして10のタイミングで投入されている。広告キャンペーンの前半に厚いのがくさび型の特徴である。広

告投入されたタイミングで正の比率が断続的に高まっていることが見て取れる(図50).

とくにステップ3あたりでアーリーマジョリティが、ステップ5あたりでレイトマジョリティがシミュレーションに参加したタイミングでプラスの比率が高まっている.

後述の短期流行型と同様に前半に注力したリーチパターンであるといえるが、広告キャンペーン期間の後半にもメディア変数が投入されるリーチパターンなので、広告キャンペーン期間中プラス比率の低下が少ないといえる.

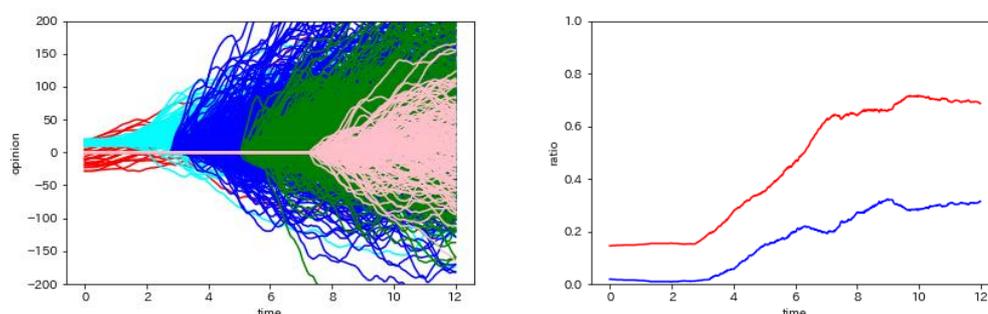


図 51 : $N = 1000$ の計算結果. 人的ネットワークはリンク接続確率 0.15 のランダムネットワーク. イノベーターの初期態度は $-30 \leq I(t) \leq 30$ で一様乱数で設定. アーリーアダプターの態度初期値は $10 \leq I(t) \leq 20$ で一様乱数で設定. その他のエージェントの態度の初期値は 0. 左図: 各採用者カテゴリー(赤: N_{inn} , 水色: N_{ea} , 青: N_{em} , 緑: N_{lm} , ピンク: N_{lg})の態度の軌跡の時間発展を提示. 右図: 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が負 (-). アーリーマジョリティとレイトマジョリティからアーリーアダプターへの信頼係数 D_{ij} は 1 から 2 の間で一様乱数によって設定, それ以外のエージェント間の信頼係数 D_{ij} は -1 から 1 の間で一様乱数によって決定. メディア効果は $(A(1, 2, 3, 6, 7, 10) = 10, A(4, 5, 8, 9, 11, 12) = 0)$. 広告の投下パターンはくさび型.

5.2.2.4 逆くさび・対人影響型によるシミュレーション

ここではマスメディア効果 $A(t) = 10$ が、ステップ 1, 4 と 5, そして 8 から 10 の時に投下されている. オピニオンリーダーと目されるアーリーアダプターの初期態度が 10 から 20 の間でセットされている点と、アーリーアダプターがアーリーマジョ

リティとレイトマジョリティから信頼されている点がここでのシミュレーションのポイントである(図 52).

左図の青(アーリーマジョリティ)と緑(レイトマジョリティ)の初期態度はゼロであるが、シミュレーション参加直後からプラスに跳ねていることが見て取れる。

シミュレーション前半では初期態度がポジティブのアーリーアダプターにメディア効果によってポジティブが強化され、後からシミュレーションに参加してくるアーリーマジョリティとレイトマジョリティに正の影響を与える。さらに広告接触による態度変容が促されている。

当該対象物に好意的なオピニオンリーダーからマジョリティに向けた影響力を期待しつつ、先行して市場にいる少数のエージェントに少量の広告をピンポイントで代入し、あとから市場(シミュレーション)に参加してくるエージェントに向けたマスメディア変数を代入するリーチパターンは効果的といえる。

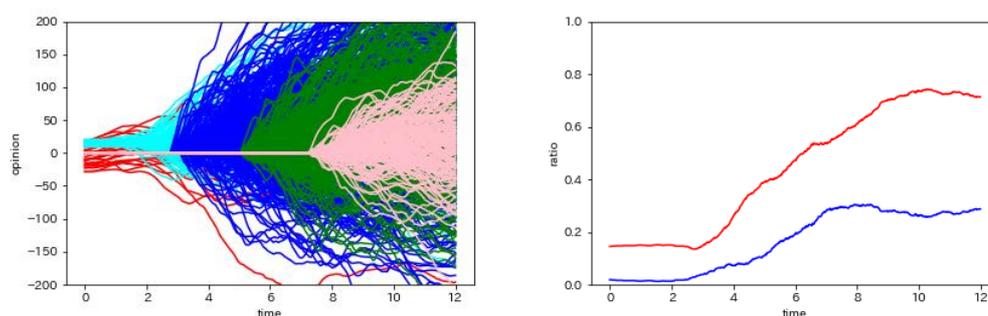


図 52 : $N = 1000$ の計算結果. 人的ネットワークはリンク接続確率 0.15 のランダムネットワーク. イノベーターの初期態度は $-30 \leq I(t) \leq 30$ で一様乱数で設定. アーリーアダプターの態度初期値は $10 \leq I(t) \leq 20$ で一様乱数で設定. その他のエージェントの態度の初期値は 0. 左図 : 各採用者カテゴリー(赤 : N_{inn} , 水色 : N_{ea} , 青 : N_{em} , 緑 : N_{lm} , ピンク : N_{lg})の態度の軌跡の時間発展を提示. 右図 : 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が不正 (-). アーリーマジョリティとレイトマジョリティからアーリーアダプターへの信頼係数 D_{ij} は 1 から 2 の間で一様乱数によって設定, それ以外のエージェント間の信頼係数 D_{ij} は -1 から 1 の間で一様乱数によって決定. メディア効果は $A(1, 4, 5, 8, 9, 10) = 10$. $A(2, 3, 6, 7, 11, 12) = 0$. 広告の投下パターンは逆くさび・対人影響型.

5.2.2.5 短期流行型によるシミュレーション

ここではマスメディア効果 $A(t) = 10$ が、ステップ1から6までの間に投下されている。大量のメディア変数が短期間に投下されているので、ポジティブの態度を持つエージェントの比率はステップ6の段階で約60%となっている。

その後メディア変数の投下がなくてもステップ9までポジティブの構成比は上昇を続けるが、その後徐々に低下していく。アーリーアダプターを信用しているアーリーマジョリティとレイトマジョリティが市場に参加するタイミングまでメディア効果が期待できる点と、アーリーアダプターがポジティブで他のエージェントに影響を与えている点が高いポジティブの構成比につながっていると考えられる(図53)。

プロダクトライフサイクルの短い流行商品を一気に大衆に認知させるのに適しているといえる。

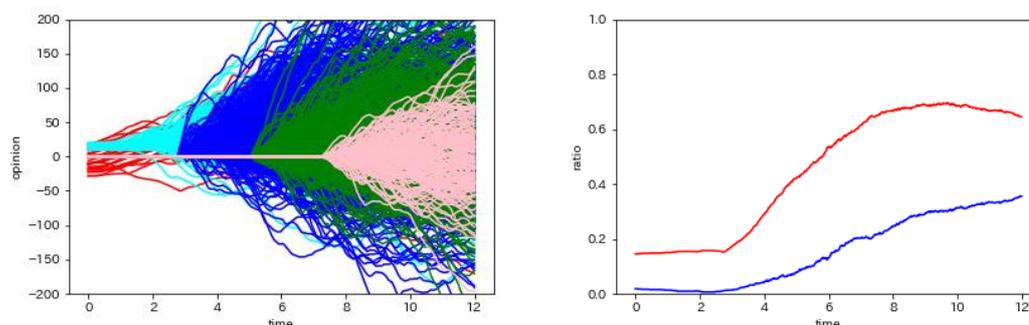


図 53: $N = 1000$ の計算結果. 人的ネットワークはリンク接続確率 0.15 のランダムネットワーク. イノベーターの初期態度は $-30 \leq I(t) \leq 30$ で一様乱数で設定. アーリーアダプターの態度初期値は $10 \leq I(t) \leq 20$ で一様乱数で設定. その他のエージェントの態度の初期値は 0. 左図: 各採用者カテゴリー(赤: N_{inn} , 水色: N_{ea} , 青: N_{em} , 緑: N_{lm} , ピンク: N_{lg})の態度の軌跡の時間発展を提示. 右図: 正 (+) と負 (-) の割合の時系列変化, 赤が正 (+), 青が不正 (-). アーリーマジョリティとレイトマジョリティからアーリーアダプターへの信頼係数 D_{ij} は 1 から 2 の間で一様乱数によって設定, それ以外のエージェント間の信頼係数 D_{ij} は -1 から 1 の間で一様乱数によって決定. メディア効果: $A(1-6)=10$. $A(7-12) = 0$. 広告の投下パターンは短期流行型.

5.2.2.6 5カテゴリーエージェントによるシミュレーションの比較

図 54 は本節で実行した各シミュレーションのうち、ポジティブの構成比の推移のみをグラフ化したものである。これによりリーチパターンごとの態度形成推移を確認することが出来る。1 エージェントのシミュレーション結果では、すべてのリーチパターンでポジティブの構成比が 97% と同一の値となったが、本章のシミュレーションではリーチパターンごとに異なる計算結果を得た(図 54)。

最終的に最も高い値を示したのは電撃型の 74.6% であった。電撃型は、全ステップにメディア変数が投下された結果、常に人々をポジティブ方向に導くことが出来たと考えられる。

2 番目にポジティブの構成比が高かったのは、逆くさび・対人影響型の 71.3% であった。逆くさび・対人影響型は、前半より後半に厚いリーチパターンなので、後半にかけてポジティブの構成比が上がったと考えられる。前半はオピニオンリーダーの影響力を期待した彼らに向けたメディア出稿で、後半はマスに向けたメディア出稿でポジティブの構成比を高める作戦は、前半でオピニオンリーダーにリーチすることが出来るなら、今の時代に適したリーチパターンと言える。

3 番目にポジティブの構成比が高かったリーチパターンは、くさび型の 67.7% であった。くさび型は、新製品が上市されるタイミング、つまり広告キャンペーンの前半にメディア出稿を厚くし、徐々にメディア出稿を減少させるリーチパターンである。メディア出稿の厚いシミュレーションの前半では、イノベーターとアーリーアダプターにしかリーチせず、アーリーマジョリティとレイトマジョリティがシミュレーションに参加するタイミングでのメディア出稿が薄くなってしまふことが、懸念材料と言える。

ポジティブの構成比が最も低い結果となったのは短期流行型の 63.3% であった。短期流行型では、前半分(1 ステップから 6 ステップまで)にのみメディア投入があり、後ろ半分にはなかった。そのため後半ではエージェント間の相互作用により、ポジテ

タイプの構成比が低下していったと考えられる。実際6ステップ目までポジティブの構成比が最も高かったのは短期流行型であった。商品ライフサイクルの短い流行品において、広告キャンペーンの前半で多くの支持を得ることが出来る短期流行型は適しているといえる。

最終結果だけで判断するのではなく、商品特性や市場状況などを考慮し、広告キャンペーンの前中後半のどこに山をもってくるのが良いのかを判断し、リーチパターンを選定することが重要である。

くさび型においても、短期流行型レベルで広告キャンペーン期間の前半にメディアを投下することが出来れば、中盤後半に投下するメディアも機能するので、十分機能すると考えられる。

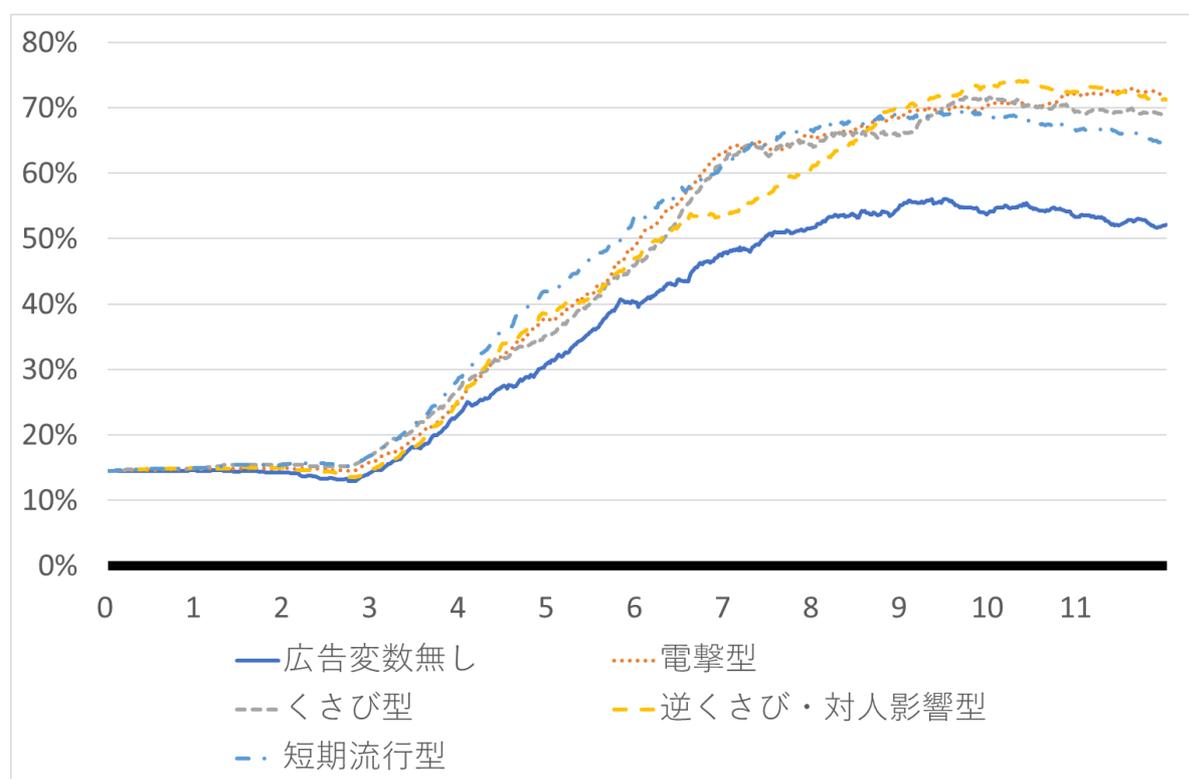


図 54: 5 カテゴリーエージェント Trust-Distrust モデルを用いたリーチパターン別ポジティブ構成比の推移

5.3 まとめ

本章は外部メディアによる影響 $A(t)$ に代表的なリーチパターンを適用し, Trust-Distrust モデルによるシミュレーションを実行した. その結果, 人々はメディアの方向に向かって態度を変容させるが, 態度変容はリーチパターンの違いによって異なることが確認できた.

またマスメディア変数が投下されていないタイミングでは, ネガティブの比率が上昇する傾向も確認できた. これは人々のイノベーションや態度へのポジティブ/ネガティブな反応は, メディアからの影響だけではなく, 人と人との接触からも影響を受けることの表れである.

本研究ではエージェントのメディア接触 (リーチ&フリークエンシー) は一律として計算したが, この先は例えばアーリーアダプターのメディア接触率は高く, ラガードのそれは低いなど, エージェントごとに調整が必要であろう. 今後はエージェントの特性に沿ったパラメータ調整を試み, より実践的なシミュレーションへとつなげていきたい.

第6章 まとめ

6.1 各章で実施したシミュレーション

表4に本研究で行ったシミュレーションをまとめている。

表4：各シミュレーションの目的

章	シミュレーション	目的
第3章	S 1	2 カテゴリーエージェントによるシミュレーション 最初から市場に興味を持っているエージェントの操作 マスメディア効果の操作 ランダムネットワークの接続確率の操作
	S 2	5 カテゴリーエージェントによるシミュレーション 5 エージェントモデルによるシミュレーション 意見の初期分布を操作したシミュレーション 信頼係数 D_{ij} の操作 マスメディア効果の操作
第4章	S 3	ネットワークの違いによるシミュレーション ランダムネットワークによるシミュレーション スケールフリーネットワークによるシミュレーション スモールワールドネットワークによるシミュレーション
	S 4	シミュレーションに広告変数を加味したシミュレーション ランダムネットワークによるシミュレーション スケールフリーネットワークによるシミュレーション スモールワールドネットワークによるシミュレーション
	S 5	5 アダプターカテゴリーによるシミュレーション ランダムネットワークによるシミュレーション スケールフリーネットワークによるシミュレーション スモールワールドネットワークによるシミュレーション
第5章	S 6	1 カテゴリーエージェントによるシミュレーション 広告変数無し 電撃型 くさび型 逆くさび・対人影響型 短期流行型
	S 7	5 カテゴリーエージェントによるシミュレーション 広告変数無し 電撃型 くさび型 逆くさび・対人影響型 短期流行型

本研究ではエージェント数、エージェントのカテゴリー数、エージェントの信頼係数、態度/意見の初期値など各種パラメータを操作した時の態度分布の違いを確認した。各種パラメータを変更することで態度分布の違いは生じるが、シミュレーションのステップ毎にエージェント間で相互作用を繰り返し、メディアの影響を受け、自身の態度を形成していく点が通底しているといえる。以下では各シミュレーションを振り返っていく。

まず S1 では、Trust-Distrust モデルのエージェントを2つのカテゴリーに拡張したシミュレーションを実施している。シミュレーションでは、エージェントの構成比の変化、マスメディア変数の調整、そしてランダムネットワークの接続確率を変化させることで、態度形成の違いについて観察している。人と人の繋がりが疎の場合には、マスメディアの影響を受けて態度が形成されるが、接続確率が高い場合、他人の態度の影響を受ける効果が大きく、メディア効果によって態度変容が起こりにくいことが示唆された。S2 ではエージェントにロジャースの採用者カテゴリーを採用した Trust-Distrust モデルによるシミュレーションを実施している。採用曲線がS字カーブを描くことが確認出来た点、またオピニオンリーダーと目されるアーリーアダプターの信頼係数を操作することで、他エージェントの態度変容に影響を与える点を確認した。S3 では1エージェントの Trust-Distrust モデルを用い、エージェント間のネットワークをランダムネットワーク、スケールフリーネットワーク、そしてスモールワールドネットワークにてシミュレーションを実施している。ネットワークの違いによって、最終的な態度の分布が異なることを観察した。S4 ではS3 にメディア変数を加えたシミュレーションを実行し、ネットワークの違いがエージェントの態度変容に与える影響について観察した。ハブ、他のエージェントに多くのリンクを持つエージェントが人々の態度変容に影響を与える点、そしてハブが情報の媒介として機能することで広告の効果がより発揮されることを確認した。これはインフルエンサー・マーケティングが効果的であることを示唆している。S5 では、5アダプターカテゴリーを適用し

た Trust-Distrust モデルにランダムネットワーク、スケールフリーネットワーク、そしてスモールワールドネットワークにてシミュレーションを実施している。最終的な態度分布は、エージェント間の相互作用がネットワークごとに異なることで違いを生んでいることを示唆した。

S6 と S7 では、メディア変数の投下パターンを、広告の実務でも用いられる 4 パターンにてシミュレーションを実施した。S6 では 1 カテゴリーエージェント、S7 では 5 アダプターカテゴリーを適用した Trust-Distrust モデルが用いられている。エージェントはメディアの方向に向かって態度を変容させるが、態度分布の変動はリーチパターンの違いによって異なることが確認された。対象とする市場の成熟度などによって広告の投下パターンを変化させることが重要である点が示唆された。

6.2 本研究の貢献

本研究ではイノベーションの普及、消費者の商品採用意向の形成過程を、オピニオンダイナミクス理論を用いて観察することを目的に、Trust-Distrust モデルのエージェント、マスメディア変数、ネットワーク構造、エージェントの初期の態度やエージェント間の信頼係数を操作しシミュレーションを実行した。エージェントのイノベーションに対する態度は、マスメディアによるバイアスは勿論、エージェント間のネットワーク構造にも影響されることが確認できた。対象とする市場構造を捉まえることで、より効率的なマーケティング活動が出来る点を示唆した。またエージェントを 5 アダプターカテゴリーに拡張させたシミュレーションでは、エージェントの採用曲線が S 字を再現している点、またマスメディア変数を実務でも用いられるリーチパターンでのシミュレーションを可能にしている点は、マーケティング、広告の領域にて実務への適用においても十分な可能性を持っているといえる。

6.3 今後の課題

本研究では Trust-Distrust モデルの基本的な変数操作によるシミュレーションに注力したため、実データを用いたシミュレーションを実行するまでには至らなかった。この点に関しては、実務で用いられた広告データの入手は容易ではないが、今後もし入手することができたら広告効果検証など実務へ大きなインプリケーションを持った研究が実施できると予想される。

またモデルの精緻化という点では、競合の存在、複数メディアの取扱い、エージェントへの広告到達率などをモデルに組み込むことが考えられる。SNS の進展により人々のインタラクションがより注目される現状から、マーケティング・コミュニケーション領域、実務での Trust-Distrust モデル活用という観点からも今後より幅広い検証が必要である。

謝辞

本研究に多大なるご指導を賜りました石井晃教授と古川勝教授に心より御礼申し上げます。

本論文は、2021年12月に急逝された石井晃教授のご指導の下で得られたオピニオンダイナミクスに関する知見に基づいて議論を展開しています。指導教官として多大な御指導を賜りました石井先生に心より御礼申し上げますとともに、心よりご冥福をお祈り申し上げます。

古川先生には、石井先生がお亡くなりになったあと、遅々として進まない論文執筆に対して暖かいご指導を賜りました。古川先生のご指導が無ければ、論文の完成はなかったかもしれません。心より感謝しております。

また石井先生がお亡くなりになられたあと、残された石井ゼミのメンバーに自主ゼミという形でご指導下さいました石井ゼミOBの松本英博先生、そして最後まで共に研究した石井ゼミのメンバー岡野のぞみさん、畑谷駿介さん、堀祐之介さんからは、多くの示唆に富むアドバイスを頂きました。この場を借りまして御礼申し上げます。また数理モデル構築と、PythonによるTrust-Distrust modelのシミュレーションに際しては、石井研究室卒業生の安部大河さんに多大なるご協力を頂きました。深く感謝します。

最後に働きながら博士課程への進学を快諾してくれた妻の元子、精神的支えとなった娘の真珠に心から感謝します。

参考文献

1. Barabasi Albert-Laszlo. *Linked: The New Science of Networks*. Perseus Books Group, 2002. (青木薫 訳『新ネットワーク思考—世界のしくみを読み解く』, HNK 出版, 2002).
2. Christakis A. Nichoas, , Fowler, H. James . *Connected*, Little. Brown and Company, 2009. (鬼澤忍 訳『つながり 社会的ネットワークの驚くべき力』, 講談社, 2010).
3. Pentland Alex. *Social Physics: How Good Ideas Spread-The Lessons from a New Science*. Penguin Books, 2014. (小林啓倫 訳『ソーシャル物理学:「良いアイデアはいかに広がるか」の新しい科学』, 草思社, 2015).
4. Solomon R. Michael. *Consumer behavior - buying, having, and being*. Pearson Education, 2013. (松井剛 監訳 『ソロモン 消費者行動論』, 丸善出版, 2015).
5. 青木幸弘, 新倉貴士; 佐々木壮太郎; 松下光司. *消費者行動論 -- マーケティングとブランド構築への応用*. 有斐閣, 2012.
6. 水野誠. *マーケティングは進化する*. 同文館出版, 2014.
7. 鳥海不二夫 編著. *計算社会科学入門*. 丸善出版, 2021.
8. A formal theory of social power. French R. P. John. 1956 年, *Psychological Review* 63(3), ページ: 181-194.
9. Harary Frank. A criterion for unanimity in French as theory of social power. CartwrightDorwin. *Studies in Social Power*. Research Center for Group Dynamics, Institute for Social Research, University of Michigan, Ann Arbor, 1959.
10. Abelson Paul Robert. *Mathematical models of the distribution of attitudes under controversy*. Norman FrederiksenGulliksenHarold. *Contributions to Mathematical Psychology*. Holt Rinehart & Winston, 1964.
11. Reaching a Consensus. DeGroot H. Morris. 345, *Journal of the American Statistical Association*, 1974 年, 第 69 巻, ページ: 118-121.
12. Social consensus and rational agnoiology. Lehrer Keith. *Synthese*, 1975 年, 第 31 巻, ページ: 141-160.
13. Reaching a consensus: Some limit theorems. Chatterjee Samprit. 1975 年, *Proceedings of International Statistical Institute*, ページ: 159-164.
14. RogersM.Everett. *Diffusion of Innovations*. Free Press, 2003 (三藤利雄 訳『イノベーションの普及』, 翔泳社, 2020).

15. <https://www.python.org/> 2022年10月10日.
16. <https://www.python.jp/index.html> 2022年10月10日.
17. <https://www.r-project.org/> 2022年10月10日.
18. Alina Sirbu, ほかに. Opinion Dynamics: Models, Extensions and External Effects. Loreto Vittorio, ほかに. Participatory Sensing, Opinions and Collective Awareness. Springer, 2016.
19. Threshold Models of Collective Behavior. Granovetter Mark. 6, The University of Chicago Press, 1978年5月, American Journal of Sociology, 第83巻, ページ: 1420-1443.
20. A Model for Spatial Conflict. Clifford Peter, Sudbury Aidan. 3, Oxford University Press, 1973年12月, Biometrika, 第60巻, ページ: 581-588.
21. Ergodic Theorems for Weakly Interacting Infinite Systems and the Voter Model. Holley A. Richard, Liggett M. Thomas. 4, Institute of Mathematical Statistics, 1975年8月, The Annals of Probability, 第3巻, ページ: 643-663.
22. Rational group decision making: A random field Ising model at $T = 0$. Galam Serge. 1-4, 1997年4月, Physica A: Statistical Mechanics and its Applications, 第238巻, ページ: 66-80.
23. The Trump phenomenon: An explanation from sociophysics. Galam Serge. 10, World Scientific Publishing Co Pte Ltd, 2017年, International Journal of Modern Physics B, 第31巻.
24. Galam Serge. Are referendums a mechanism to turn our prejudices into rational choices? An unfortunate answer from sociophysics. Morel Laurence, Qvortrup Matt. The Routledge Handbook to Referendums and Direct Democracy. London: Taylor & Francis, 2018, 19.
25. Uniformity, Bipolarization and Pluriformity Captured as Generic Stylized Behavior with an Agent-Based Simulation Model of Attitude Change. Jager Wander, Amblard Frédéric. 2005年, Computational and Mathematical Organization Theory, 第10巻, ページ: 295-303.
26. Multiple attitude dynamics in large populations. Jager Wander, Amblard Frédéric. Argonne National Laboratory, The University of Chicago, 2005年1月, Conference Paper: Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms.
27. Dynamics of bounded confidence opinion in heterogeneous social networks: Concord against partial antagonism. Kurmyshev Evguenii, Juárez A. Héctor, González-Silva A. Ricardo. 16, 2011年8月, Physica A: Statistical Mechanics and its Applications, 第390巻, ページ: 2945-2955.

28. Mixing Beliefs among Interacting Agents. Deffuant Guillaume, ほか. 01n04, World Scientific, 2000 年 1 月, Advances in Complex Systems, 第 3 巻, ページ: 87-98.
29. Meet, Discuss and Segregate! Weisbuch Gérard, Deffuant Guillaume Amblard Frédéric; Nadal Jean-Pierre. 3, Hindawi, 2002 年 1 月, Complexity, 第 7 巻, ページ: 55 - 63.
30. Opinion dynamics and bounded confidence: models, analysis and simulation. Hegselmann Rainer , KrauseUlrich. 3, Journal of Artificial Societies and Social Simulation, 2002 年 7 月, Journal of Artificial Societies and Social Simulation, 第 5 巻.
31. Ishii Akira , Kawahata Yasuko. Opinion Dynamics Theory for Analysis of Consensus Formation and Division of Opinion on the Internet. 2018. Proceedings of the 22nd Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES2018).
32. Ishii Akira. Opinion Dynamics Theory Considering Trust and Suspicion in Human Relations. Morais Costa Danielle, Costa Carreras, Ashley de Almeida, Adiel Teixeira; Vetschera, Rudolf. Group Decision and Negotiation: Behavior, Models, and Support. 19th International Conference, GDN 2019, Loughborough, UK, June 11-15, 2019, Proceedings. Springer, 2019, 第 351 巻, ページ: 193-204.
33. Keller Ed , Fay Brad. The Face-to-Face Book. Free Press, 2012 (澁谷覚・久保田進彦・須永努訳『フェイス・トゥ・フェイス・ブック -- クチコミ・マーケティングの効果を最大限に高める秘訣』有斐閣, 2016) .
34. 安田由紀. ネットワーク分析 何が行為を決定するのか. 新曜社, 1997.
35. 坂口英継 , 本庄春雄. 複雑系科学への招待. サイエンス社, 2018.
36. BarabásiAlbert-László. Cambridge University Press, 2016 (池田裕一, 井上寛康, 谷澤俊弘 訳『ネットワーク科学: ひと・もの・ことの関係性をデータから解き明かす新しいアプローチ』, 共立出版, 2019).
37. Fass Craig, Ginelli Mike , Turtle Brian. Six Degrees of Kevin Bacon. Plume Books, 1996.
38. ReynoldsPatrick. the Oracle of Bacon. (オンライン) 1999 年. (引用日: 2022 年 10 月 10 日.) <https://oracleofbacon.org/>.
39. KotlerPhilip , KellerLaneKevin. Marketing Management. Pearson, 2005 (恩蔵直人 監訳 『コトラー&ケラーのマーケティング・マネジメント 第 12 版』, 丸善出版, 2014) .
40. Bandwagon, Snob and Veblen Effect in the Theory of Consumers' Demand. Leibenstein Harvey. 2, Oxford University Press, 1950 年 5 月, The Quarterly Journal of Economics, 第 64 巻, ページ: 183-207 .

41. 古川一郎, 守口剛, 阿部誠. マーケティング・サイエンス入門. 有斐閣アルマ.
42. A New Product Growth for Model Consumer Durables. Bass M. Frank. 2, 1969年, Management Science, 第 15 巻, ページ: 215-227.
43. 片平秀貴. マーケティング・サイエンス. 東京大学出版会, 1994.
44. 朝野熙彦. 入門多変量解析の実際. 講談社サイエンティフィック, 2009.
45. 朝野熙彦. 最新マーケティング・サイエンスの基礎. 講談社, 2010.
46. 馬場真哉. 時系列分析と状態空間モデルの基礎: R と Stan で学ぶ理論と実装. プレアデス出版, 2018.
47. 沖本竜義. 経済・ファイナンスデータの計量時系列分析. 朝倉書店, 2010.
48. Campagnoli Patrizia, Petrone Sonia, Petris Giovanni. Dynamic Linear Models with R. Springer, 2009.
(和合 肇(監訳)／萩原 淳一郎(訳)『Rによる ベイジアン動的線型モデル』, 朝倉書店, 2013) , .
49. 樋口知之. 予測にいかす統計モデリングの基本—ベイズ統計入門から応用まで. 講談社, 2011.
50. 中妻照雄. Python による ベイズ統計学入門. 朝倉書店, 2019.
51. 朝野熙彦, 土田尚弘; 河原達也; 藤居誠. ビジネスマンが一步先をめざす ベイズ統計学 — Excel から RStan へステップアップ. 2018.
52. Fujii Makoto, Ishii Akira. The simulation of diffusion of innovations using new opinion dynamics. The 5th International Workshop on Application of Big Data for Computational Social Science(ABCSS2020 @ WI-IAT 2020), 2020.
53. 松井剛, 西川英彦 (編著) . 1からの消費者行動. 碩学舎, 2016.
54. 岸志津江, 田中洋, 嶋村和江. 現代広告論』. 有斐閣, 2017.
55. W.シュラム (編集) 学習院大学社会学研究室(翻訳). マス・コミュニケーション—マス・メディアの総合的研究. 東京創元社, 1968.
56. GiepFranzen. Advertising Effectiveness: Findings from Empirical Research. 出版地不明 : NTC Publications, 1994(八巻俊雄, 嶋村和江, 丸岡吉仁 訳 『広告効果 データと理論からの再検証』, 日本経済新聞社, 1996) .
57. RossiterR.John, BellmanSteven. Marketingu Cimmunications: Theory and Applications. Pearson-Prentice Hall, 2005 (岸志津江 監訳, 『戦略的マーケティング・コミュニケーション IMC の理論と実際』, 東急エージェンシー, 2009) .

付録

Trust-Distrust モデル・ソースコード

```
1 #Trust-Distrust モデル2カテゴリー - ランダムネットワーク版
2
3
4 #!/usr/bin/env python3
5 # -*- coding: utf-8 -*-
6 """
7 Created on Fri Mar 15 10:03:41 2019
8 """
9 # 正の分布と負の分布に分けて計算をスタート
10 # 正の分布と負の分布はそれぞれ+20, -20 を中心に幅 20
11
12 # 正の分布のグループと負の分布のグループで別々に Dij を設定
13
14 # カリスマを入れた計算
15 # カリスマは0番目と1番目
16 # カリスマの意志を人々は変更(10 倍に設定)
17 # 人々からのカリスマへの信頼を 5.0 に設定
18
19 # ダミーを含めて N 人として, あとから興味を持って加わってくる
20 # 150 人が最初から興味を持つ
21
22 # 150 人はあとから興味を持つ
23 # 追加される 150 人の興味の持ち方は  $1/2(\tanh(x)-1)$  で単純に設定
24 # 関数 AddUser() で設定している
25
26 from scipy.integrate import odeint
27 import numpy as np
```

```

28 import math
29 import matplotlib.pyplot as plt
30 import matplotlib.animation as animation
31 import csv
32 import random
33 import pandas as pd
34
35 n = 300
36 Nini = 30
37
38 #最初から興味を持つ人数
39 Nuser = Nini
40
41 # 宣伝
42 a = 1  #* np.random.rand(1)
43 #num = random.randint(1, 100)
44 #if num > 50:
45 #    a = 10
46 #else:
47 #    a = 0
48
49 #乱数シート固定
50 #np.random.seed(123)
51
52 d = np.random.rand(n, n) * 2 - 1
53 o = np.random.rand(n) * 2 - 1
54
55 # 態度の初期分布の設定
56 for k in range(0, Nini):

```

```
57 # o[k] = ((np.random.rand()*2-1)*30
58     o[k] = (np.random.rand()*(30)
59 # o[k] = (np.random.rand()*(-30)
60
61 for k in range(Nini+1, n):
62     o[k] = 0.0
63
64 # グループ内, グループ間の信頼 Dij の設定
65 for i in range(0, Nini):
66     for j in range(0, Nini):
67         d[i, j] = np.random.rand()*2-1
68
69 for i in range(Nini+1, n):
70     for j in range(Nini+1, n):
71         d[i, j] = np.random.rand()*2-1
72
73 for i in range(0, Nini):
74     for j in range(Nini+1, n):
75         d[i, j] = np.random.rand()*2-1
76
77 for i in range(Nini+1, n):
78     for j in range(0, Nini):
79         d[i, j] = np.random.rand()*2-1
80         d[i, j] = np.random.rand()
81 # d[i, j] = np.random.rand()*(-1)
82
83 # random netowok 設定
84 # 指定の数が, ランダムネットワークで各ノードからリンクが繋がっている割合
85 for i in range(0, n):
```

```

86     for j in range(0, n):
87         if np.random.rand() > 0.5:
88             d[i, j] = 0
89
90     # o[0] = 15 # カリスマの人 A の態度の初期値の設定
91     # o[299] = -15 # カリスマの人Bの態度の初期値の設定
92
93     # カリスマへの人々からの信頼
94     #for k in range(n):
95     #    d[k, 0]=5.0
96     #    d[k, 299]=5.0
97     # d[1, 0]= 1.0 #1の人の0の人へば信頼度の設定
98
99     positive = []
100    negative = []
101    zero = []
102
103    tmin = 0
104    tmax = 10
105    z = 500
106    h = (tmax-tmin)/z
107
108    high = 10
109
110    # mass
111    mass = []
112    for k in range(n):
113        mass.append(1.0)
114

```

```
115 # カリスマの人の意志を決める
116 #mass[0] = 10.
117 #mass[299] = 10.
118
119
120 I = 0
121 e = np.array(n)
122 t = tmin
123 time = np.array([])
124 time = np.append( time, np.array([t]))
125
126 def Phi(x, y, beta, alpha, t):
127     if (beta*abs(y-x) - alpha) > 10:
128         return 0
129     else:
130         return 1
131
132 beta = 1
133 alpha = 10
134
135 def AddUser(t):
136     return (math.tanh(t-5)+1)/2
137
138
139 while (t <= tmax):
140     Nuser = Nini + int((n - Nini)*AddUser(t)) #興味を持っている人の数
141     for i in range(Nuser):
142         b = 0
143         for j in range(Nuser):
```

```

144         if j==i:
145             b = b + d[i, i] * o[i]
146         else:
147             b = b + d[i, j] * Link[i, j] * (o[j]-o[i]) * Phi(o[j], o[i], beta, alpha,
148 t)/mass[i]
149 #             b = b + d[i, j] * (o[j]-o[i]) * Phi(o[j], o[i], beta, alpha, t)/mass[i]
150         e = np.append( e, np.array([ a + b - d[i, i] * o[i]]))
151     for i in range(Nuser+1, n):
152         e = np.append(e, 0)
153
154     e = e * h
155     o = o + e
156     I = np.vstack((I, o))
157     t += h
158     time = np.append( time, np.array([t]))
159     size = time.size
160     e = 0
161     size = 0
162     e = np.array(n)
163
164     size = time.size
165
166     plt.figure()
167     for k in range(1, Nini):
168         plt.plot(time, I[:, k], color='pink')
169
170     for k in range(Nini+1, n):
171         plt.plot(time, I[:, k], color='cyan')
172

```

```
173 #plt.plot(time, I[:, 0], color='blue')
174 #plt.plot(time, I[:, 299], color='red')
175 # plt.plot(time, I[:, 2], color='green')
176 #highline = np.full(size, high)
177 #plt.plot(time, highline, color='blue')
178 #lowline = np.full(size, -high)
179 #plt.plot(time, lowline, color='blue')
180
181 plt.xlabel("time")
182 plt.ylabel("opinion")
183 plt.xlim()
184 plt.ylim(-200, 200)
185 plt.savefig('n=' + str(n) + '_sw30_ikn+only_Nini=T_a1_1.png')
186
187 plt.figure()
188 plt.hist(I[size-1, :], bins = 160, color = 'skyblue', range = (-200, 200), histtype
189 = 'bar')
190 plt.xlabel("opinion")
191 plt.ylabel("distribution")
192 plt.xlim(-40, 40)
193 plt.ylim(0, 30)
194 plt.savefig('n=' + str(n) + '_sw30_ikn+only_Nini=T_a1_2.png')
195
196 for t in range(0, z):
197     posi = 0
198     nega = 0
199     zero = 0
200     for k in range(0, n):
201         if I[t, k] >= 0.001:
```

```

202         posi = posi+1
203     elif I[t, k] < -0.001:
204         nega = nega+1
205     else:
206         zero = zero+1
207     positive.append(posi/n)
208     negative.append(nega/n)
209
210 plt.figure()
211 plt.plot(time[0:z], positive, color='red')
212 plt.plot(time[0:z], negative, color='blue')
213 plt.xlabel("time")
214 plt.ylabel("ratio")
215 plt.xlim()
216 plt.ylim(0, 1)
217 plt.savefig('n=' + str(n) + '_sw30_ikn+only_Nini=T_a1_3.png')
218
219 #print(positive)
220 #print(negative)
221 #print(zero)
222
223 positive = positive
224 negative = negative
225 dict = {'positive': positive, 'negative': negative}
226 df = pd.DataFrame(dict)
227 df.to_csv('sw30_ikn+only_Nini=T_a1.csv')
228

```

229

```
1 #Trust-Distrust モデル5カテゴリー - ランダムネットワーク版
2
3 #!/usr/bin/env python3
4 # -*- coding: utf-8 -*-
5 """
6 Created on Fri Mar 15 10:03:41 2019
7
8 """
9 # 正の分布と負の分布に分けて計算をスタート
10 # 正の分布と負の分布はそれぞれ+20, -20 を中心に幅 20
11
12 # 正の分布のグループと負の分布のグループで別々に Dij を設定
13
14 # カリスマを入れた計算
15 # カリスマは0番目と1番目
16 # カリスマの意志を人々は変更(10 倍に設定)
17 # 人々からのカリスマへの信頼を 5.0 に設定
18
19 # ダミーを含めて N 人として, あとから興味を持って加わってくる
20 # 150 人が最初から興味を持つ
21
22 # 150 人はあとから興味を持つ
23 # 追加される 150 人の興味の持ち方は  $1/2(\tanh(x)-1)$  で単純に設定
24 # 関数 AddUser() で設定している
25
26 from scipy.integrate import odeint
27 import numpy as np
28 import math
```

```

29 import random
30 import matplotlib.pyplot as plt
31 import matplotlib.animation as animation
32 import csv
33
34 n = 1000
35 Ninn = 25
36 Nea = 135
37 Nem = 340
38 Nlm = 340
39 Nlg = 160
40
41 Ninnea = Ninn + Nea
42 Ninneaem = Ninn + Nea + Nem
43 Ninneaemlm = Ninn + Nea + Nem + Nlm
44 Ninneaemlmlg = Ninn + Nea + Nem + Nlm + Nlg #=1000
45
46
47 #最初から興味を持つ人数
48 Nuser = Ninn
49
50 # 宣伝
51 a = 0    #* np.random.rand(1)
52
53 #乱数シート固定
54 np.random.seed(123)
55
56 d = np.random.rand(n, n) * 2 - 1
57 o = np.random.rand(n) * 2 - 1

```

```
58
59 # 態度の初期分布の設定, イノベータをプラスのみにすることも可能
60 for k in range(0, Ninn):
61     o[k] = ((np.random.rand()*2-1)*30
62     #o[k] = ((np.random.rand()*30)-20)
63
64 for k in range(Ninn+1, Ninn + Nea):
65     o[k] = 0.0
66     #o[k] = ((np.random.rand()*2-1)*10
67     #o[k] = ((np.random.rand()*10)+10)
68     #o[k] = ((np.random.rand()*(-10)+(-10)))
69
70 for k in range(Ninn+Nea+1, Ninn + Nea + Nem):
71     o[k] = 0.0
72
73 for k in range(Ninn+Nea+Nem+1, Ninn + Nea + Nem + Nlm):
74     o[k] = 0.0
75
76 for k in range(Ninn+Nea+Nem+Nlm+1, Ninn + Nea + Nem + Nlm + Nlg):
77     o[k] = 0.0
78
79
80 # グループ内, グループ間の信頼 Dij の設定, 25 この関係になる
81
82 #inn
83 for i in range(0, Ninn):
84     for j in range(0, Ninn):
85         d[i, j] = np.random.rand()*2-1
86 for i in range(0, Ninn):
```

```

87     for j in range(Ninn+1, Ninnea):
88         d[i, j] = np.random.rand()*2-1
89 for i in range(0, Ninn):
90     for j in range(Ninneae+1, Ninneaem):
91         d[i, j] = np.random.rand()*2-1
92 for i in range(0, Ninn):
93     for j in range(Ninneaeem+1, Ninneaemlm):
94         d[i, j] = np.random.rand()*2-1
95 for i in range(0, Ninn):
96     for j in range(Ninneaeemlm +1, 1000):
97         d[i, j] = np.random.rand()*2-1
98
99
100 #ea
101 for i in range(Ninn+1, Ninnea):
102     for j in range(0, Ninn):
103         d[i, j] = np.random.rand()*2-1
104 for i in range(Ninn+1, Ninnea):
105     for j in range(Ninn+1, Ninnea):
106         d[i, j] = np.random.rand()*2-1
107         #d[i, j] = random.uniform((-0.5), 1.0)
108         #d[i, j] = random.uniform(1.0, 2.0)
109         #d[i, j] = np.random.rand()
110 for i in range(Ninn+1, Ninnea):
111     for j in range(Ninneae+1, Ninneaem):
112         d[i, j] = np.random.rand()*2-1
113 for i in range(Ninn+1, Ninnea):
114     for j in range(Ninneaeem+1, Ninneaemlm):
115         d[i, j] = np.random.rand()*2-1

```

```
116 for i in range(Ninn+1, Ninnea):
117     for j in range(Ninneae/mlm +1, 1000):
118         d[i, j] = np.random.rand()*2-1
119
120
121 #em
122 for i in range(Ninneae+1, Ninneaem):
123     for j in range(0, Ninn):
124         d[i, j] = np.random.rand()*2-1
125 for i in range(Ninneae+1, Ninneaem):
126     for j in range(Ninn+1, Ninnea):
127         d[i, j] = np.random.rand()*2-1
128         #d[i, j] = random.uniform((-0.5), 1.0)
129         #d[i, j] = random.uniform(1.0, 2.0)
130         #d[i, j] = np.random.rand()
131 for i in range(Ninneae+1, Ninneaem):
132     for j in range(Ninneae+1, Ninneaem):
133         d[i, j] = np.random.rand()*2-1
134 for i in range(Ninneae+1, Ninneaem):
135     for j in range(Ninneaeem+1, Ninneaem/mlm):
136         d[i, j] = np.random.rand()*2-1
137 for i in range(Ninneae+1, Ninneaem):
138     for j in range(Ninneae/mlm +1, 1000):
139         d[i, j] = np.random.rand()*2-1
140
141
142 #lm
143 for i in range(Ninneaeem+1, Ninneaem/mlm):
144     for j in range(0, Ninn):
```

```

145         d[i, j] = np.random.rand()*2-1
146     for i in range(Ninneaem+1, Ninneaemlm):
147         for j in range(Ninn+1, Ninnea):
148             d[i, j] = np.random.rand()*2-1
149             #d[i, j] = random.uniform((-0.5), 1.0)
150             #d[i, j] = random.uniform(1.0, 2.0)
151             #d[i, j] = np.random.rand()
152     for i in range(Ninneaem+1, Ninneaemlm):
153         for j in range(Ninnea+1, Ninneaem):
154             d[i, j] = np.random.rand()*2-1
155     for i in range(Ninneaem+1, Ninneaemlm):
156         for j in range(Ninneaem+1, Ninneaemlm):
157             d[i, j] = np.random.rand()*2-1
158     for i in range(Ninneaem+1, Ninneaemlm):
159         for j in range(Ninneaemlm +1, 1000):
160             d[i, j] = np.random.rand()*2-1
161
162
163     #lg
164     for i in range(Ninneaemlm +1, 1000):
165         for j in range(0, Ninn):
166             d[i, j] = np.random.rand()*2-1
167     for i in range(Ninneaemlm +1, 1000):
168         for j in range(Ninn+1, Ninnea):
169             d[i, j] = np.random.rand()*2-1
170             #d[i, j] = random.uniform((-0.5), 1.0)
171             #d[i, j] = random.uniform(1.0, 2.0)
172             #d[i, j] = np.random.rand()
173     for i in range(Ninneaemlm +1, 1000):

```

```
174     for j in range(Ninneam+1, Ninneaelm):
175         d[i, j] = np.random.rand()*2-1
176 for i in range(Ninneaelm +1, 1000):
177     for j in range(Ninneaelm+1, Ninneaelm):
178         d[i, j] = np.random.rand()*2-1
179 for i in range(Ninneaelm +1, 1000):
180     for j in range(Ninneaelm +1, 1000):
181         d[i, j] = np.random.rand()*2-1
182
183
184 # random network 設定
185 # 指定の数が, ランダムネットワークで各ノードからリンクが繋がっている割合
186 for i in range(0, n):
187     for j in range(0, n):
188         if np.random.rand() > 0.50:
189             d[i, j] = 0
190
191
192 # o[0] = 15 # カリスマの人 A の態度の初期値の設定
193 # o[299] = -15 # カリスマの人Bの態度の初期値の設定
194
195 # カリスマへの人々からの信頼
196 #for k in range(n):
197 #    d[k, 0]=5.0
198 #    d[k, 299]=5.0
199 #    d[1, 0]= 1.0 #1の人の0の人へば信頼度の設定
200
201 positive = []
202 negative = []
```

```

203
204 tmin = 0
205 tmax = 15
206 z = 500
207 h = (tmax-tmin)/z
208
209 high = 10
210
211 # mass
212 mass = []
213 for k in range(n):
214     mass.append(1.0)
215
216 # カリスマの人の意志を決める
217 #mass[0] = 10.
218 #mass[299] = 10.
219
220
221 I = 0
222 e = np.array(n)
223 t = tmin
224 time = np.array([])
225 time = np.append( time, np.array([t]))
226
227 def Phi(x, y, beta, alpha, t):
228     if (beta*abs(y-x) - alpha) > 10:
229         return 0
230     else:
231         return 1

```

```
232
233 beta = 1
234 alpha = 10
235
236 #def AddUser(t):
237     #return (math.tanh(t-5)+1)/2
238
239 def Finn(t):
240     return (math.tanh(t)+1)/2
241
242 def Fea(t):
243     return (math.tanh(t-3)+1)/2
244
245 def Fem(t):
246     return (math.tanh(t-6)+1)/2
247
248 def Flm(t):
249     return (math.tanh(t-9)+1)/2
250
251 def Flg(t):
252     return (math.tanh(t-12)+1)/2
253
254
255 while (t <= tmax):
256     #Nuser = Nini + int((n - Nini)*AddUser(t)) #興味を持っている人の数
257     Nuser = Ninn*Finn(t) + Nea*Fea(t) + Nem*Fem(t) + Nlm*Flm(t) + Nlg*Flg(t) #
258     順次加入各関数を考える
259     for i in range(int(Nuser)):
260         b = 0
```

```

261     for j in range(int(Nuser)):
262         if j==i:
263             b = b + d[i, i] * o[i]
264 #         elif o[i]-o[j]>=high or o[i]-o[j]<=-high:
265 #             b = b
266         else:
267             b = b + d[i, j] * (o[j]-o[i]) * Phi(o[j], o[i], beta, alpha, t)/mass[i]
268         e = np.append( e, np.array([ a + b - d[i, i] * o[i]]))
269     for i in range(int(Nuser)+1, n):
270         e = np.append(e, 0)
271
272     e = e * h
273     o = o + e
274     I = np.vstack((I, o))
275     t += h
276     time = np.append( time, np.array([t]))
277     size = time.size
278     e = 0
279     size = 0
280     e = np.array(n)
281
282     size = time.size
283
284     plt.figure()
285     for k in range(1, Ninn):
286         plt.plot(time, I[:, k], color='red')
287
288     for k in range(Ninn+1, Ninnea):
289         plt.plot(time, I[:, k], color='cyan')

```

```
290
291 for k in range(Ninnea+1, Ninneaem):
292     plt.plot(time, I[:, k], color='blue')
293
294 for k in range(Ninneaem+1, Ninneaemlm):
295     plt.plot(time, I[:, k], color='green')
296
297 for k in range(Ninneaemlm+1, 1000):
298     plt.plot(time, I[:, k], color='pink')
299
300 #koko
301 #plt.plot(time, I[:, 0], color='blue')
302 #plt.plot(time, I[:, 299], color='red')
303 #plt.plot(time, I[:, 2], color='green')
304 #highline = np.full(size, high)
305 #plt.plot(time, highline, color='blue')
306 #lowline = np.full(size, -high)
307 #plt.plot(time, lowline, color='blue')
308 #koko
309
310 plt.xlabel("time")
311 plt.ylabel("opinion")
312 plt.xlim()
313 plt.ylim(-200, 200)
314 plt.savefig('n=' + str(n) + '_1.png')
315
316 plt.figure()
317 plt.hist(I[size-1, :], bins = 160, color = 'skyblue', range = (-200, 200), histtype
318 = 'bar')
```

```

319 plt.xlabel("opinion")
320 plt.ylabel("distribution")
321 plt.xlim(-40, 40)
322 plt.ylim(0, 30)
323 plt.savefig('n=' + str(n) + '_2.png')
324
325 for t in range(0, z):
326     posi = 0
327     nega = 0
328     zero = 0
329     for k in range(0, n):
330         if I[t, k] >= 0.001:
331             posi = posi+1
332         elif I[t, k] < -0.001:
333             nega = nega+1
334         else:
335             zero = zero+1
336     positive.append(posi/n)
337     negative.append(nega/n)
338
339 plt.figure()
340 plt.plot(time[0:500], positive, color='red')
341 plt.plot(time[0:500], negative, color='blue')
342 plt.xlabel("time")
343 plt.ylabel("ratio")
344 plt.xlim()
345 plt.ylim(0, 1)
346 plt.savefig('n=' + str(n) + '_3.png')

```

Trust-Distrust モデルのシミュレーション_スケールフリーネットワーク, スモールワールドネットワーク版

```
1 #ネットワーク作成～Trust-Distrust モデルへの組込み
2
3 #次数に応じた描画
4 import networkx as nx
5 import numpy as np
6 import matplotlib.pyplot as plt
7 n=1000
8 #m=1
9
10 #スケールフリーをつくる
11 #G = nx.barabasi_albert_graph(n, m)
12 #G = nx.barabasi_albert_graph(n = 100, m=1)
13 #スモールワールドをつくる(n はノード数, k は次数, p はつながり変え確率)
14 G = nx.watts_strogatz_graph(n, k = 4, p = 0.1, seed = 1)
15
16 #図表の作成. figsize は図表の大きさ
17 #plt.figure(figsize=(20, 15))
18 plt.subplots(figsize=(6, 6))
19
20 #図表のレイアウトを決める. k の値が小さい程図表が密集する
21 #pos = nx.random_layout(G)
22 #pos=nx.spring_layout(G)#初期 pos
23 pos = nx.circular_layout(G)
24
25 #ノードとエッジの描画
26 # _color: 色の指定
27 # alpha: 透明度の指定
28 #nx.draw_networkx_edges(G, pos, edge_color='y', width=1)#edge の描画
29 #nx.draw_networkx_nodes(G, pos, node_color='r', alpha=0.5)
30 nx.draw_networkx(G, pos, node_color="w", edgecolors='k', node_size=100,
31 with_labels=False)
32 nx.draw_networkx_labels(G, pos, font_size=10, font_family='DejaVu Sans',
```

```

33  dpi=2000)
34
35  #ノードラベルを付加
36  #nx.draw_networkx_labels(G, pos, font_size=10)
37  #nx.draw_networkx_labels(G, pos, font_size=15)
38
39  #X 軸 Y 軸を表示しない設定
40  plt.axis('off')
41
42  #PageRank の追加(次数が大きいものほど大きく表示)
43  #pr = nx.degree_centrality(G)
44  #node_size に PageRank の値を組み込む
45  #nx.draw_networkx_nodes(G , pos , node_color='b' , alpha=0.5 ,
46  node_size=[5000*v for v in pr.values()])#node の描画
47  #nx.draw_networkx_labels(G , pos , font_size=10 , font_family='DejaVu
48  Sans', dpi=2000)
49  plt.savefig("sw300.png")
50
51  #図表を描画
52  plt.show(G)
53
54  #エッジ
55  G.edges()
56
57
58  #ディグリー
59  G.degree()
60
61
62  ary = nx.to_numpy_array(G, dtype=int)
63  print(ary)
64
65
66  import csv
67  #Link

```

```
68 print("Link を保存中")
69 #f = open('complete_net, Link.csv', 'w')
70 f = open('sf1000_1', 'w')
71 w = csv.writer(f, lineterminator = '¥n')
72 column=[]
73 for i in range(n):
74     for j in range(n):
75         column= np.append( column, float(ary[i, j]))
76     w.writerow(column)
77     column = []
78 f.close()
79
80
81 #Link 読み込み
82 #file = open(str(number) + ', Link.csv', 'r')
83 #file = open('complete_net, Link.csv', 'r')
84 #file = open('random_net, 0.5, Link.csv', 'r')
85 file = open('sf1000_1', 'r')
86 reader = csv.reader(file)
87 data00 = [row for row in reader]
88 data_Link = np.array(data00)
89 #data = data0[1:, 1:]
90 file.close()
91
92
93 #リンク
94 Link = np.random.rand(n, n)
95 for i in range(n):
96     for j in range(n):
97         Link[i, j] = float(data_Link[i, j]) #int だとエラー
98
99
100 #作成したネットワークの Trust-Distrust モデルへの組込みとシミュレー
101 ション
102
```

```

103  #!/usr/bin/env python3
104  # -*- coding: utf-8 -*-
105  """
106  Created on Fri Mar 15 10:03:41 2019
107  """
108  # 正の分布と負の分布に分けて計算をスタート
109  # 正の分布と負の分布はそれぞれ+20, -20 を中心に幅 20
110
111  # 正の分布のグループと負の分布のグループで別々に Dij を設定
112
113  # カリスマを入れた計算
114  # カリスマは0番目と1番目
115  # カリスマの意志を人々は変更(10 倍に設定)
116  # 人々からのカリスマへの信頼を 5.0 に設定
117
118
119  from scipy.integrate import odeint
120  import numpy as np
121  import math
122  import random
123  import matplotlib.pyplot as plt
124  import matplotlib.animation as animation
125  import csv
126  import pandas as pd
127
128
129  n = 1000
130  Ninn = 25
131  Nea = 135
132  Nem = 340
133  Nlm = 340
134  Nlg = 160
135
136  Ninnea = Ninn + Nea
137  Ninneaem = Ninn + Nea + Nem

```

```
138 Ninneaemlm = Ninn + Nea + Nem + Nlm
139 Ninneaemlmlg = Ninn + Nea + Nem + Nlm + Nlg #=1000
140
141
142 #最初から興味を持つ人数
143 Nuser = Ninn
144
145 # 宣伝
146 a = 0    #* np.random.rand(1)
147
148 #乱数シート固定
149 np.random.seed(123)
150
151 d = np.random.rand(n, n) * 2 - 1
152 o = np.random.rand(n) * 2 - 1
153
154 # 態度の初期分布の設定, イノベータをプラスのみにすることも可能
155 for k in range(0, Ninn):
156     o[k] = ((np.random.rand()*2-1)*30
157     #o[k] = ((np.random.rand()*30)-20)
158
159 for k in range(Ninn+1, Ninn + Nea):
160     o[k] = 0.0
161     #o[k] = ((np.random.rand()*2-1)*10
162     #o[k] = ((np.random.rand()*10)+10)
163     #o[k] = ((np.random.rand()*(-10)+(-10)))
164
165 for k in range(Ninn+Nea+1, Ninn + Nea + Nem):
166     o[k] = 0.0
167
168 for k in range(Ninn+Nea+Nem+1, Ninn + Nea + Nem + Nlm):
169     o[k] = 0.0
170
171 for k in range(Ninn+Nea+Nem+Nlm+1, Ninn + Nea + Nem + Nlm + Nlg):
172     o[k] = 0.0
```

```

173
174
175 # グループ内, グループ間の信頼 Dij の設定, 25 この関係になる
176
177 #inn
178 for i in range(0, Ninn):
179     for j in range(0, Ninn):
180         d[i, j] = np.random.rand()*2-1
181 for i in range(0, Ninn):
182     for j in range(Ninn+1, Ninnea):
183         d[i, j] = np.random.rand()*2-1
184 for i in range(0, Ninn):
185     for j in range(Ninneam+1, Ninneaem):
186         d[i, j] = np.random.rand()*2-1
187 for i in range(0, Ninn):
188     for j in range(Ninneam+1, Ninneaemlm):
189         d[i, j] = np.random.rand()*2-1
190 for i in range(0, Ninn):
191     for j in range(Ninneamlm +1, 1000):
192         d[i, j] = np.random.rand()*2-1
193
194
195 #ea
196 for i in range(Ninn+1, Ninnea):
197     for j in range(0, Ninn):
198         d[i, j] = np.random.rand()*2-1
199 for i in range(Ninn+1, Ninnea):
200     for j in range(Ninn+1, Ninnea):
201         d[i, j] = np.random.rand()*2-1
202         #d[i, j] = random.uniform((-0.5), 1.0)
203         #d[i, j] = random.uniform(1.0, 2.0)
204         #d[i, j] = np.random.rand()
205 for i in range(Ninn+1, Ninnea):
206     for j in range(Ninneam+1, Ninneaem):
207         d[i, j] = np.random.rand()*2-1

```

```
208 for i in range(Ninn+1, Ninnea):
209     for j in range(Ninneae+1, Ninneaemlm):
210         d[i, j] = np.random.rand()*2-1
211 for i in range(Ninn+1, Ninnea):
212     for j in range(Ninneae+1, 1000):
213         d[i, j] = np.random.rand()*2-1
214
215
216 #em
217 for i in range(Ninneae+1, Ninneaem):
218     for j in range(0, Ninn):
219         d[i, j] = np.random.rand()*2-1
220 for i in range(Ninneae+1, Ninneaem):
221     for j in range(Ninn+1, Ninnea):
222         d[i, j] = np.random.rand()*2-1
223         #d[i, j] = random.uniform((-0.5), 1.0)
224         #d[i, j] = random.uniform(1.0, 2.0)
225         #d[i, j] = np.random.rand()
226 for i in range(Ninneae+1, Ninneaem):
227     for j in range(Ninneae+1, Ninneaem):
228         d[i, j] = np.random.rand()*2-1
229 for i in range(Ninneae+1, Ninneaem):
230     for j in range(Ninneae+1, Ninneaemlm):
231         d[i, j] = np.random.rand()*2-1
232 for i in range(Ninneae+1, Ninneaem):
233     for j in range(Ninneae+1, 1000):
234         d[i, j] = np.random.rand()*2-1
235
236
237 #lm
238 for i in range(Ninneae+1, Ninneaemlm):
239     for j in range(0, Ninn):
240         d[i, j] = np.random.rand()*2-1
241 for i in range(Ninneae+1, Ninneaemlm):
242     for j in range(Ninn+1, Ninnea):
```

```

243     d[i, j] = np.random.rand()*2-1
244     #d[i, j] = random.uniform((-0.5), 1.0)
245     #d[i, j] = random.uniform(1.0, 2.0)
246     #d[i, j] = np.random.rand()
247 for i in range(Ninneaem+1, Ninneaemlm):
248     for j in range(Ninneaem+1, Ninneaem):
249         d[i, j] = np.random.rand()*2-1
250 for i in range(Ninneaem+1, Ninneaemlm):
251     for j in range(Ninneaem+1, Ninneaemlm):
252         d[i, j] = np.random.rand()*2-1
253 for i in range(Ninneaem+1, Ninneaemlm):
254     for j in range(Ninneaemlm +1, 1000):
255         d[i, j] = np.random.rand()*2-1
256
257
258 #lg
259 for i in range(Ninneaemlm +1, 1000):
260     for j in range(0, Ninn):
261         d[i, j] = np.random.rand()*2-1
262 for i in range(Ninneaemlm +1, 1000):
263     for j in range(Ninn+1, Ninnea):
264         d[i, j] = np.random.rand()*2-1
265         #d[i, j] = random.uniform((-0.5), 1.0)
266         #d[i, j] = random.uniform(1.0, 2.0)
267         #d[i, j] = np.random.rand()
268 for i in range(Ninneaemlm +1, 1000):
269     for j in range(Ninneaem+1, Ninneaem):
270         d[i, j] = np.random.rand()*2-1
271 for i in range(Ninneaemlm +1, 1000):
272     for j in range(Ninneaem+1, Ninneaemlm):
273         d[i, j] = np.random.rand()*2-1
274 for i in range(Ninneaemlm +1, 1000):
275     for j in range(Ninneaemlm +1, 1000):
276         d[i, j] = np.random.rand()*2-1
277

```

```
278
279 # random netowok 設定
280 # 指定の数が, ランダムネットワークで各ノードからリンクが繋がっている割合
281 #for i in range(0, n):
282 #    for j in range(0, n):
283 #        if np.random.rand() > 0.5:
284 #            d[i, j] = 0
285 #ネットワーク
286 #Link = np.ones((n, n))#行列
287 Link=Link
288 #print("L=", Link)
289
290
291 # o[0] = 15 # カリスマの人 A の態度の初期値の設定
292 # o[299] = -15 # カリスマの人Bの態度の初期値の設定
293
294 # カリスマへの人々からの信頼
295 #for k in range(n):
296 #    d[k, 0]=5.0
297 #    d[k, 299]=5.0
298 #    d[1, 0]= 1.0 #1の人の0の人へば信頼度の設定
299
300 positive = []
301 negative = []
302
303 tmin = 0
304 tmax = 15
305 z = 500
306 h = (tmax-tmin)/z
307
308 high = 10
309
310 # mass
311 mass = []
312 for k in range(n):
```

```

313     mass.append(1.0)
314
315     # カリスマの人の意志を決める
316     #mass[0] = 10.
317     #mass[299] = 10.
318
319
320     I = o
321     e = np.array(n)
322     t = tmin
323     time = np.array([])
324     time = np.append( time, np.array([t]))
325
326     def Phi(x, y, beta, alpha, t):
327         if (beta*abs(y-x) - alpha) > 10:
328             return 0
329         else:
330             return 1
331
332     beta = 1
333     alpha = 10
334
335     #def AddUser(t):
336         #return (math.tanh(t-5)+1)/2
337
338     def Finn(t):
339         return (math.tanh(t)+1)/2
340
341     def Fea(t):
342         return (math.tanh(t-3)+1)/2
343
344     def Fem(t):
345         return (math.tanh(t-6)+1)/2
346
347     def Flm(t):

```

```

348     return (math.tanh(t-9)+1)/2
349
350 def Flg(t):
351     return (math.tanh(t-12)+1)/2
352
353
354 while (t <= tmax):
355     #Nuser = Nini + int((n - Nini)*AddUser(t)) #興味を持っている人の数
356     Nuser = Ninn*Finn(t) + Nea*Fea(t) + Nem*Fem(t) + Nlm*Flm(t) +
357     Nlg*Flg(t) #順次加入各関数を考える
358     for i in range(int(Nuser)):
359         b = 0
360         for j in range(int(Nuser)):
361             if j==i:
362                 b = b + d[i, i] * o[i]
363             # elif o[i]-o[j]>=high or o[i]-o[j]<=-high:
364             #     b = b
365             else:
366                 b = b + d[i, j] * Link[i, j] * (o[j]-o[i]) * Phi(o[j], o[i], beta, alpha,
367 t)/mass[i]
368             #     b = b + d[i, j] * (o[j]-o[i]) * Phi(o[j], o[i], beta, alpha,
369 t)/mass[i]
370         e = np.append( e, np.array([ a + b - d[i, i] * o[i]]))
371     for i in range(int(Nuser)+1, n):
372         e = np.append(e, 0)
373
374     e = e * h
375     o = o + e
376     I = np.vstack((I, o))
377     t += h
378     time = np.append( time, np.array([t]))
379     size = time.size
380     e = 0
381     size = 0
382     e = np.array(n)

```

```

383
384 size = time.size
385
386 plt.figure()
387 for k in range(1, Ninn):
388     plt.plot(time, I[:, k], color='red')
389
390 for k in range(Ninn+1, Ninnea):
391     plt.plot(time, I[:, k], color='cyan')
392
393 for k in range(Ninnea+1, Ninneaem):
394     plt.plot(time, I[:, k], color='blue')
395
396 for k in range(Ninneaem+1, Ninneaemlm):
397     plt.plot(time, I[:, k], color='green')
398
399 for k in range(Ninneaemlm+1, 1000):
400     plt.plot(time, I[:, k], color='pink')
401
402 #koko
403 #plt.plot(time, I[:, 0], color='blue')
404 #plt.plot(time, I[:, 299], color='red')
405 #plt.plot(time, I[:, 2], color='green')
406 #highline = np.full(size, high)
407 #plt.plot(time, highline, color='blue')
408 #lowline = np.full(size, -high)
409 #plt.plot(time, lowline, color='blue')
410 #koko
411
412 plt.xlabel("time")
413 plt.ylabel("opinion")
414 plt.xlim()
415 plt.ylim(-200, 200)
416 plt.savefig('n=' + str(n) + '_sf5cn1000_a0_1.png')
417

```

```
418 plt.figure()
419 plt.hist(I[size-1, :], bins = 160, color = 'skyblue', range = (-200, 200),
420 histtype = 'bar')
421 plt.xlabel("opinion")
422 plt.ylabel("distribution")
423 plt.xlim(-40, 40)
424 plt.ylim()
425 plt.savefig('n=' + str(n) + '_sf5cn1000_a0_2.png')
426
427 for t in range(0, z):
428     posi = 0
429     nega = 0
430     zero = 0
431     for k in range(0, n):
432         if I[t, k] >= 0.001:
433             posi = posi+1
434         elif I[t, k] < -0.001:
435             nega = nega+1
436         else:
437             zero = zero+1
438     positive.append(posi/n)
439     negative.append(nega/n)
440
441 plt.figure()
442 plt.plot(time[0:500], positive, color='red')
443 plt.plot(time[0:500], negative, color='blue')
444 plt.xlabel("time")
445 plt.ylabel("ratio")
446 plt.xlim()
447 plt.ylim(0, 1)
448 plt.savefig('n=' + str(n) + '_sf5cn1000_a0_3.png')
449
450 #print(positive)
451 #print(negative)
452 #print(zero)
```

```

453
454 positive = positive
455 negative = negative
456 dict = {'positive': positive, 'negative': negative}
457 df = pd.DataFrame(dict)
458 df.to_csv('sf5cn1000_a0.csv')
459
460
461 # Trust-Distrust モデル_外生変数(リーチパターン)操作版
462 #!/usr/bin/env python3
463 # -*- coding: utf-8 -*-
464 # 正の分布と負の分布に分けて計算をスタート
465 # 正の分布と負の分布はそれぞれ+20, -20 を中心に幅 20
466
467 # 正の分布のグループと負の分布のグループで別々に Dij を設定
468
469 # カリスマを入れた計算
470 # カリスマは 0 番目と 1 番目
471 # カリスマの意志を人々は変更 (10 倍に設定)
472 # 人々からのカリスマへの信頼を 5.0 に設定
473
474 from scipy.integrate import odeint
475 import numpy as np
476 import math
477 import random
478 import matplotlib.pyplot as plt
479 import matplotlib.animation as animation
480 import csv
481 import pandas as pd
482
483 n = 1000
484 Ninn = 25
485 Nea = 135
486 Nem = 340
487 Nlm = 340
488 Nlg = 160

```

```
489
490 Ninnea = Ninn + Nea
491 Ninneaem = Ninn + Nea + Nem
492 Ninneaemlm = Ninn + Nea + Nem + Nlm
493 Ninneaemlmlg = Ninn + Nea + Nem + Nlm + Nlg #=1000
494
495 #最初から興味を持つ人数
496 Nuser = Ninn
497
498 # 宣伝
499 #a = 0
500 #a = 5
501 a = 10
502
503 #乱数シート固定
504 np.random.seed(123)
505
506 d = np.random.rand(n, n) * 2 - 1
507 o = np.random.rand(n) * 2 - 1
508
509 # 態度の初期分布の設定, イノベータをプラスのみにすることも可能
510 for k in range(0, Ninn):
511     o[k] = ((np.random.rand()*2-1)*30
512     #o[k] = ((np.random.rand()*30)-20)
513
514 for k in range(Ninn+1, Ninn + Nea):
515     #o[k] = 0.0
516     #o[k] = ((np.random.rand()*2-1)*10
517     o[k] = ((np.random.rand()*10)+10)
518     #o[k] = ((np.random.rand()*(-10)+(-10)))
519
520 for k in range(Ninn+Nea+1, Ninn + Nea + Nem):
521     o[k] = 0.0
522
523 for k in range(Ninn+Nea+Nem+1, Ninn + Nea + Nem + Nlm):
524     o[k] = 0.0
```

```

525
526 for k in range(Ninn+Nea+Nem+Nlm+1, Ninn + Nea + Nem + Nlm + Nlg):
527     o[k] = 0.0
528
529
530 # グループ内, グループ間の信頼 Dij の設定, 25 この関係になる
531
532 #inn
533 for i in range(0, Ninn):
534     for j in range(0, Ninn):
535         d[i, j] = np.random.rand()*2-1
536 for i in range(0, Ninn):
537     for j in range(Ninn+1, Ninnea):
538         d[i, j] = np.random.rand()*2-1
539 for i in range(0, Ninn):
540     for j in range(Ninnea+1, Ninneaem):
541         d[i, j] = np.random.rand()*2-1
542 for i in range(0, Ninn):
543     for j in range(Ninneaem+1, Ninneaemlm):
544         d[i, j] = np.random.rand()*2-1
545 for i in range(0, Ninn):
546     for j in range(Ninneaemlm + 1, 1000):
547         d[i, j] = np.random.rand()*2-1
548
549
550 #ea
551 for i in range(Ninn+1, Ninnea):
552     for j in range(0, Ninn):
553         d[i, j] = np.random.rand()*2-1
554 for i in range(Ninn+1, Ninnea):
555     for j in range(Ninn+1, Ninnea):
556         #d[i, j] = np.random.rand()*2-1
557         d[i, j] = random.uniform((-0.5), 1.0)
558         #d[i, j] = random.uniform(1.0, 2.0)
559         #d[i, j] = np.random.rand()

```

```
560 for i in range(Ninn+1, Ninnea):
561     for j in range(Ninneae+1, Ninneaem):
562         d[i, j] = np.random.rand()*2-1
563 for i in range(Ninn+1, Ninnea):
564     for j in range(Ninneae+1, Ninneaemlm):
565         d[i, j] = np.random.rand()*2-1
566 for i in range(Ninn+1, Ninnea):
567     for j in range(Ninneae+1, 1000):
568         d[i, j] = np.random.rand()*2-1
569
570
571 #em
572 for i in range(Ninneae+1, Ninneaem):
573     for j in range(0, Ninn):
574         d[i, j] = np.random.rand()*2-1
575 for i in range(Ninneae+1, Ninneaem):
576     for j in range(Ninn+1, Ninnea):
577         #d[i, j] = np.random.rand()*2-1
578         #d[i, j] = random.uniform((-0.5), 1.0)
579         d[i, j] = random.uniform(1.0, 2.0)
580         #d[i, j] = np.random.rand()
581 for i in range(Ninneae+1, Ninneaem):
582     for j in range(Ninneae+1, Ninneaem):
583         d[i, j] = np.random.rand()*2-1
584 for i in range(Ninneae+1, Ninneaem):
585     for j in range(Ninneae+1, Ninneaemlm):
586         d[i, j] = np.random.rand()*2-1
587 for i in range(Ninneae+1, Ninneaem):
588     for j in range(Ninneae+1, 1000):
589         d[i, j] = np.random.rand()*2-1
590
591
592 #lm
593 for i in range(Ninneae+1, Ninneaemlm):
594     for j in range(0, Ninn):
```

```

595         d[i, j] = np.random.rand()*2-1
596     for i in range(Ninneae+1, Ninnea+1):
597         for j in range(Ninn+1, Ninnea):
598             #d[i, j] = np.random.rand()*2-1
599             #d[i, j] = random.uniform((-0.5), 1.0)
600             d[i, j] = random.uniform(1.0, 2.0)
601             #d[i, j] = np.random.rand()
602     for i in range(Ninneae+1, Ninnea+1):
603         for j in range(Ninneae+1, Ninnea):
604             d[i, j] = np.random.rand()*2-1
605     for i in range(Ninneae+1, Ninnea+1):
606         for j in range(Ninneae+1, Ninnea+1):
607             d[i, j] = np.random.rand()*2-1
608     for i in range(Ninneae+1, Ninnea+1):
609         for j in range(Ninneae+1, 1000):
610             d[i, j] = np.random.rand()*2-1
611
612
613     #lg
614     for i in range(Ninneae+1, 1000):
615         for j in range(0, Ninn):
616             d[i, j] = np.random.rand()*2-1
617     for i in range(Ninneae+1, 1000):
618         for j in range(Ninn+1, Ninnea):
619             d[i, j] = np.random.rand()*2-1
620             #d[i, j] = random.uniform((-0.5), 1.0)
621             #d[i, j] = random.uniform(1.0, 2.0)
622             #d[i, j] = np.random.rand()
623     for i in range(Ninneae+1, 1000):
624         for j in range(Ninneae+1, Ninnea):
625             d[i, j] = np.random.rand()*2-1
626     for i in range(Ninneae+1, 1000):
627         for j in range(Ninneae+1, Ninnea+1):
628             d[i, j] = np.random.rand()*2-1
629     for i in range(Ninneae+1, 1000):

```

```
630     for j in range(Ninneae/mlm + 1, 1000):
631         d[i, j] = np.random.rand()*2-1
632
633
634 # random netowok 設定
635 # 指定の数が, ランダムネットワークで各ノードからリンクが繋がっている割合
636 for i in range(0, n):
637     for j in range(0, n):
638         if np.random.rand() > 0.15:
639             d[i, j] = 0
640
641
642 # o[0] = 15 # カリスマの人 A の態度の初期値の設定
643 # o[299] = -15 # カリスマの人 B の態度の初期値の設定
644
645 # カリスマへの人々からの信頼
646 #for k in range(n):
647 #    d[k, 0]=5.0
648 #    d[k, 299]=5.0
649 #    d[1, 0]= 1.0 # 1 の人の 0 の人へば信頼度の設定
650
651 positive = []
652 negative = []
653
654 tmin = 0
655 tmax = 12
656 z = 720
657 h = (tmax-tmin)/z
658
659 high = 10
660
661 # mass
662 mass = []
663 for k in range(n):
664     mass.append(1.0)
665
```

```

666 # カリスマの人の意志を決める
667 #mass[0] = 10.
668 #mass[299] = 10.
669
670
671 I = 0
672 e = np.array(n)
673 t = tmin
674 time = np.array([])
675 time = np.append( time,  np.array([t]))
676
677 def Phi(x, y, beta, alpha, t):
678     if (beta*abs(y-x) - alpha) > 10:
679         return 0
680     else:
681         return 1
682
683 beta = 1
684 alpha = 10
685
686 #def AddUser(t):
687     #return (math.tanh(t-5)+1)/2
688
689 def Finn(t):
690     return (math.tanh(t)+1)/2
691
692 def Fea(t):
693     return (math.tanh(t-2)+1)/2
694
695 def Fem(t):
696     return (math.tanh(t-4)+1)/2
697
698 def Flm(t):
699     return (math.tanh(t-6)+1)/2
700
701 def Flg(t):

```

```

702     return (math.tanh(t-8)+1)/2
703
704 while (t <= tmax):
705     #Nuser = Nini + int((n - Nini)*AddUser(t)) #興味を持っている人の数
706     Nuser = Ninn*Finn(t) + Nea*Fea(t) + Nem*Fem(t) + Nlm*Flm(t) + Nlg*Flg(t) #順次加入各関
707     数を考える
708     for i in range(int(Nuser)):
709         b = 0
710         for j in range(int(Nuser)):
711             if j==i:
712                 b = b + d[i, i] * o[i]
713             # elif o[i]-o[j]>=high or o[i]-o[j]<=-high:
714             #     b = b
715             else:
716                 b = b + d[i, j] * (o[j]-o[i]) * Phi(o[j], o[i], beta, alpha, t)/mass[i]
717         e = np.append( e,  np.array([ a + b - d[i, i] * o[i]]))
718         # a = (10 *np.sin(np.radians(t)))
719         # a = (-10 *np.sin(np.radians(t)))
720         # a = (10 *np.sin(np.radians(t) * 60))
721         # a = (-10 *np.sin(np.radians(t) * 60))
722         # a = (5/6)*t
723         # a = (-5/6)*t + 10
724     # kusabi 3-2-1
725     # if 0 <= t < 1:
726     #     a = 10
727     # elif 1 <= t < 2:
728     #     a = 10
729     # elif 2 <= t < 3:
730     #     a = 10
731     # elif 3 <= t < 4:
732     #     a = 0
733     # elif 4 <= t < 5:
734     #     a = 0
735     # elif 5 <= t < 6:
736     #     a = 10
737     # elif 6 <= t < 7:

```

```
738 #             a = 10
739 #     elif 7 <= t < 8:
740 #             a = 0
741 #     elif 8 <= t < 9:
742 #             a = 0
743 #     elif 9 <= t < 10:
744 #             a = 10
745 #     elif 10 <= t < 11:
746 #             a = 0
747 #     elif 11 <= t < 12:
748 #             a = 0
749 #     elif 12 <= t:
750 #             a = 0
751 # Gyaku-kusabi 1-2-3
752 #     if 0 <= t < 1:
753 #             a = 10
754 #     elif 1 <= t < 2:
755 #             a = 0
756 #     elif 2 <= t < 3:
757 #             a = 0
758 #     elif 3 <= t < 4:
759 #             a = 10
760 #     elif 4 <= t < 5:
761 #             a = 10
762 #     elif 5 <= t < 6:
763 #             a = 0
764 #     elif 6 <= t < 7:
765 #             a = 0
766 #     elif 7 <= t < 8:
767 #             a = 10
768 #     elif 8 <= t < 9:
769 #             a = 10
770 #     elif 9 <= t < 10:
771 #             a = 10
772 #     elif 10 <= t < 11:
773 #             a = 0
```

```
774 #     elif 11 <= t <12:
775 #         a = 0
776 #     elif 12 <= t:
777 #         a = 0
778 # tanki-ryuko
779     if 0 <= t < 1:
780         a = 10
781     elif 1 <= t < 2:
782         a = 10
783     elif 2 <= t < 3:
784         a = 10
785     elif 3 <= t < 4:
786         a = 10
787     elif 4 <= t < 5:
788         a = 10
789     elif 5 <= t < 6:
790         a = 10
791     elif 6 <= t < 7:
792         a = 0
793     elif 7 <= t < 8:
794         a = 0
795     elif 8 <= t < 9:
796         a = 0
797     elif 9 <= t < 10:
798         a = 0
799     elif 10 <= t < 11:
800         a = 0
801     elif 11 <= t <12:
802         a = 0
803     elif 12 <= t:
804         a = 0
805 # 定数テスト
806 #     if 0 <= t < 1:
807 #         a = 5
808 #     elif 1 <= t < 2:
809 #         a = 5
```

```

810 #     elif 2 <= t < 3:
811 #         a = 5
812 #     elif 3 <= t < 4:
813 #         a = 5
814 #     elif 4 <= t < 5:
815 #         a = 5
816 #     elif 5 <= t < 6:
817 #         a = 5
818 #     elif 6 <= t < 7:
819 #         a = 5
820 #     elif 7 <= t < 8:
821 #         a = 5
822 #     elif 8 <= t < 9:
823 #         a = 5
824 #     elif 9 <= t < 10:
825 #         a = 5
826 #     elif 10 <= t < 11:
827 #         a = 5
828 #     elif 11 <= t < 12:
829 #         a = 5
830 #     elif 12 <= t:
831 #         a = 0
832
833     for i in range(int(Nuser)+1, n):
834         e = np.append(e, 0)
835
836         e = e * h
837         o = o + e
838         I = np.vstack((I, o))
839         t += h
840         time = np.append( time, np.array([t]))
841         size = time.size
842         e = 0
843         size = 0
844         e = np.array(n)
845

```

```
846 size = time.size
847
848 plt.figure()
849 for k in range(1, Ninn):
850     plt.plot(time, I[:, k], color='red')
851
852 for k in range(Ninn+1, Ninnea):
853     plt.plot(time, I[:, k], color='cyan')
854
855 for k in range(Ninnea+1, Ninneaem):
856     plt.plot(time, I[:, k], color='blue')
857
858 for k in range(Ninneaem+1, Ninneaemlm):
859     plt.plot(time, I[:, k], color='green')
860
861 for k in range(Ninneaemlm+1, 1000):
862     plt.plot(time, I[:, k], color='pink')
863
864 #koko
865 #plt.plot(time, I[:, 0], color='blue')
866 #plt.plot(time, I[:, 299], color='red')
867 #plt.plot(time, I[:, 2], color='green')
868 #highline = np.full(size, high)
869 #plt.plot(time, highline, color='blue')
870 #lowline = np.full(size, -high)
871 #plt.plot(time, lowline, color='blue')
872 #koko
873
874 plt.xlabel("time")
875 plt.ylabel("opinion")
876 plt.xlim()
877 plt.ylim(-200, 200)
878 plt.savefig('n=' + str(n) + '_MF-B1b1.png')
879
880 plt.figure()
```

```

881 plt.hist(I[size-1, :], bins = 160, color = 'skyblue', range = (-200, 200), histtype = 'bar')
882 plt.xlabel("opinion")
883 plt.ylabel("distribution")
884 plt.xlim(-100, 100)
885 plt.ylim(0, 30)
886 plt.savefig('n=' + str(n) + '_MF-B1b2.png')
887
888 for t in range(0, z):
889     posi = 0
890     nega = 0
891     zero = 0
892     for k in range(0, n):
893         if I[t, k] >= 0.001:
894             posi = posi+1
895         elif I[t, k] < -0.001:
896             nega = nega+1
897         else:
898             zero = zero+1
899     positive.append(posi/n)
900     negative.append(nega/n)
901
902 plt.figure()
903 plt.plot(time[0:z], positive, color='red')
904 plt.plot(time[0:z], negative, color='blue')
905 plt.xlabel("time")
906 plt.ylabel("ratio")
907 plt.xlim()
908 plt.ylim(0, 1)
909 plt.savefig('n=' + str(n) + '_MF-B1b3.png')
910
911 #print(positive)
912 #print(negative)
913 #print(zero)
914 print(a)
915
916 positive = positive

```

```
917 negative = negative
918 dict = {'positive': positive, 'negative': negative}
919 df = pd.DataFrame(dict)
920 df.to_csv('_MF-B1.csv')
```