

矛盾-無関心を考慮したトラスト値の遷移について

河辺 義信¹ 小泉 佑揮² 大木 哲史³ 西垣 正勝³ 長谷川 享² 小田 哲久⁴

概要: 本稿では、信頼と不信の2つの指標に基づくトラスト値の表現法について述べる。この表現法では、メッセージや参加者に対しての「信用しているが、信用できない」といった状況を扱える。こうした矛盾した状況は、たとえば「メッセージの内容は一貫しているものの、そのメッセージは正当な送信者から送られていない」といった場合などに起こり得る。さらに本稿では、このような矛盾を許すトラスト値の時系列的な遷移に関する性質について、I/O-オートマトン理論に基づく議論を行う。

キーワード: トラスト, FCR 法, ファジィ理論, I/O オートマトン, 安全性・活性

On Trust State under Contradiction-Irrelevance and Its Transitions

YOSHINOBU KAWABE¹ YUKI KOIZUMI² TETSUSHI OHKI³ MASAKATSU NISHIGAKI³ TORU HASEGAWA²
TETSUHISA ODA⁴

Abstract: This study discusses a representation of trust values based on two metrics by trust and distrust. By our representation, we can deal with situations where you may trust and distrust messages/participants simultaneously; for example, you may trust and distrust a message if the content of the message is consistent but the message is not sent from a correct sender. We also discuss properties based on the transitions of trust values with the I/O-automaton theory.

Keywords: on-line trust, FCR method, fuzzy logic, I/O-automaton, trace safety/liveness properties

1. はじめに

近年の大規模災害においては、被災者の安否情報や救援情報の交換にソーシャルメディアが積極的に活用され、早期の安否確認や人命救助などに役立てられている。しかし災害時に交換される情報は、すべてが信頼できるとは限らない。たとえば、2018年6月18日の朝に大阪府北部を

襲った震度6弱の地震の際には、その当日中にツイッター上に「シマウマが脱走した」「京セラドーム大阪の屋根に亀裂が入っている」「京阪電車が脱線している」などの真偽不明の情報（実際は誤り）が流され、さらにリツイートされて広範囲に拡散するという事態が起こった [1]。この事例では、全くのたまたまな情報が交換されたり、情報の一部が正確でも別の部分に重大な誤りが混入していたり、過去の類似の情報を脚色し流布している。

信頼できるかわからない情報の別の例としては、時間の経過とともに情報の真偽が変わるケースが挙げられる。たとえば、「ある人は、大怪我をしているが生きている」という情報が大規模災害の発生直後に真であったとしても、その1時間後にもなお正しいとは限らない。このほかにも、「届いたメッセージの内容は、一見矛盾なく正しいと認められるものの、そのメッセージは正当な送信者から送られておらず、全く見たことのないメールアドレスから届いた」

¹ 愛知工業大学 情報科学部
Department of Information Science, Aichi Institute of Technology

² 大阪大学大学院
Graduate School of Information Science and Technology, Osaka University

³ 静岡大学大学院
Graduate School of Science and Technology, Shizuoka University

⁴ 愛知工業大学 経営学部
Department of Business Administration, Aichi Institute of Technology

という場合にも、その情報を本当に信じて良いのか、疑わしく思われることもあるだろう。

こうしたソーシャルネットワーク上で交換される情報を扱う際には、情報がどれだけ信じられるのか、また情報の発信元や受け取り先のユーザはどれだけ信じてよいのか、といった、トラスト（信頼）の評価が重要となる。Marsh と Dibben は、人や事柄に関するトラストを -1 から 1 の間の評価値として与えた。さらに、トラストに関連する概念を、トラスト（信頼, trust）、ディストラスト（不信, distrust）、アントラスト（無信頼, untrust）、およびミストラスト（誤信頼, mistrust）に分類している [8]。

Marsh と Dibben による分類は、トラストとディストラストを対極の概念とする一次元的なトラスト値の定義に基づいている。しかし一方で、トラストとディストラストを異なる次元で議論すべきとの指摘もある [5]。とくに本稿では、トラストが人の抱く印象に関連する概念であることを重要視し、その評価にあたっては「矛盾・混乱（信頼しているものの、実は同時に不信感も抱いている）」や「無関心（信頼の気持ちも不信の気持ちもない）」が考慮されるべきであるとする。そこで本稿では、印象形成に関する理論である小田のファジィ多項目並列評定法（fuzzy-set concurrent rating method. 以下、FCR 法）[12], [14], [15], [16] の考え方にに基づき、信頼の度合いと不信の度合いの組による、二次元的なトラスト値の表現を導入する。さらに、我々のトラスト値から Marsh と Dibben によるトラスト値への対応を示す。

二次元的なトラスト値の表現を状態とみなしたとき、その状態遷移の列に関する性質を考えることができる。本稿では、時系列的なトラスト値の変化に関する分析を行うため、状態遷移機械に基づくモデル化を行い、観測点の変化に関する性質を検討する。「ディストラストの状態に至らない」「いちどディストラスト以外の状態に達した後、再びディストラストの状態に至ることはない」といったトラストに関する安全性（一般に「悪い振舞いが発生しない」ことを表す）や「いつかトラストの状態に達する」などの活性（一般に「望まれる振舞いが、いつか最終的に発生する」ことを表す）について議論する。とくに安全性に関しては、トレース包含に基づく効率的な証明法についても述べる。

2. 準備

2.1 ファジィ多項目並列評定法（FCR 法）

アンケートでは「良い」「悪い」「どちらとも言えない」などの複数の項目から一つを選ぶ形式の評定尺度法（図 1）がよく用いられる。しかしこの手法には、尺度の中間項目に回答が集中するという問題点がある。たとえば、被験者が複数の選択項目に回答したい場合を考えよう。とくに回答したい項目が両極にあるとき、被験者はどちらか一方を無理矢理選ぶか、もしくは該当しないはずの中間項目を選

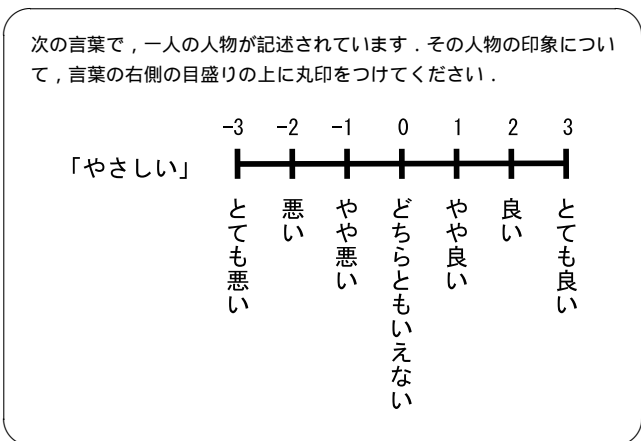


図 1 7 段階法（従来のアンケート方法）の評定例

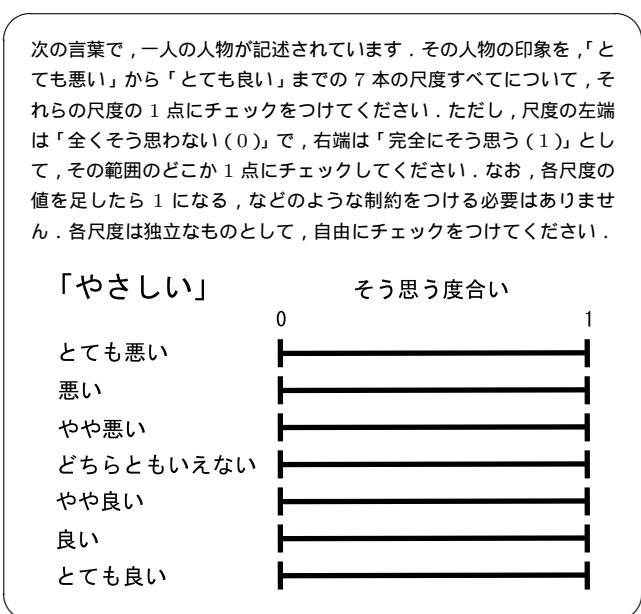


図 2 FCR 法に基づく評定の例

ぶことになる。多くの場合、中間項目は「ふつう」「どちらとも言えない」のようにラベル化されることが多く、応答結果のみからでは、両極に該当する場合であると判定できない。そのほか、質問事項に被験者が関心を持っていない場合や知識不足等で何と答えればよいかわからない場合にも、中間項目に回答する傾向が見られる。こうした中間項目への応答集中は、それ以上の分析を困難とする。

これに対して FCR 法では、それらの全項目に対して「そう思う度合い」、すなわち確信度（該当度）を評定尺度法等で測定し（図 2）、ファジィ推論を適用してそれらの答えを一つの値に集約する。これにより、従来の評定尺度法で測定できない種類の情報（たとえば、対象に対し「良いとも悪いとも思っている」といった情報）を取り出すことができる。なお、FCR 法の理論面において次元（評価尺度の数）に制限はないが、応用面からは、2 または 3 程度の次元が実用的とされる（たとえば、ある項目について「そう

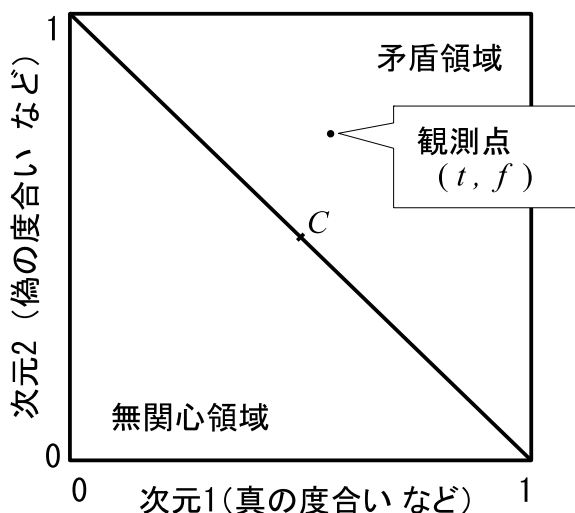


図3 2次元 HLS モデル

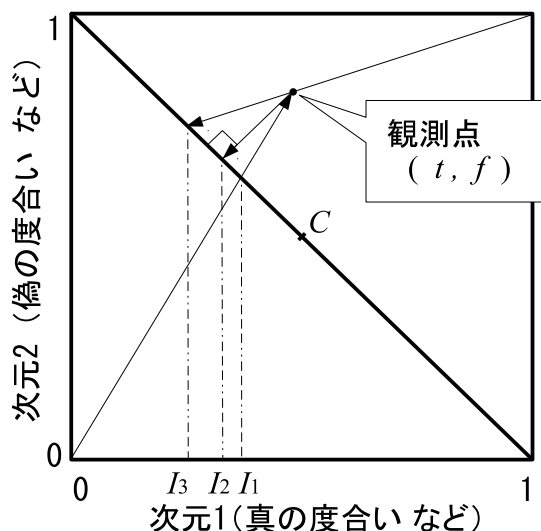


図4 統合値の計算

思う」「そう思わない」の2次元を設定するなど). 以下の説明では, 2次元の場合で述べる.

2.1.1 HLS モデル

複数の評定尺度の値からファジィ推論を行うために, FCR 法では, 多次元多値論理空間としての HLS モデル (Hyper Logic Space model) が用いられる. たとえば, 図3は, 「真(横軸. 以下 T)」と「偽(縦軸. 以下 F)」による2次元空間を表している. また, 観測点(真理値)は点 $(t, f) \in T \times F$ として表される. ここで $t, f \in [0, 1]$ であり, t と f は独立である. すなわち, 定和尺度 ($t+f=1$) などの制約は用いない. 集合 $\{(t, f) | t, f \in [0, 1] \wedge t+f > 1\}$ を矛盾領域と, 集合 $\{(t, f) | t, f \in [0, 1] \wedge t+f < 1\}$ を無関心領域 もしくは 無関連領域と呼ぶ. また, 集合 $\{(t, f) | t, f \in [0, 1] \wedge t+f = 1\}$ を, 数値真理値空間 あるいは 無矛盾領域と呼ぶ. 観測点 $A = (t_A, f_A)$ および $B = (t_B, f_B)$ を真理値とみなす場合については, HLS における論理演算が

$$\begin{aligned} A \vee B &= (\max(t_A, t_B), \min(f_A, f_B)) \\ A \wedge B &= (\min(t_A, t_B), \max(f_A, f_B)) \\ \neg A &= (1 - t_A, 1 - f_A) \end{aligned}$$

で与えられる. 論理演算の詳細については, 文献 [13] を参照されたい.

2.1.2 統合値

HLS 上の観測点 (t, f) が与えられたとき, 並列的に与えられた尺度 (t および f) から「統合値」と呼ぶ代表的な値が算出される. 統合値の計算方法は何種類かが考案されており, 解析対象に応じて使い分けられる. 主なものを以下に示す.

- 単純スコア法: t に1点, f に0点を割り当て, 重み付け平均を計算する. ただし, 分母が0のときは別に定める. 統合値 I_1 は

$$I_1(t, f) = \begin{cases} 0.5 & t+f=0 \text{ のとき} \\ \frac{t}{t+f} & \text{それ以外} \end{cases}$$

で与えられる.

- 逆転項目平均法: 否定的尺度 f の否定を $1-f$ とし, 肯定的尺度 t との平均値をとる. すなわち統合値 I_2 を

$$I_2(t, f) = \frac{t + (1 - f)}{2}$$

と定める.

- 逆スコア法: 肯定的尺度 t の否定 $1-t$ と否定的尺度 f の否定 $1-f$ に対して, 単純スコア法の式を適用する. すなわち, 統合値 I_3 は

$$I_3(t, f) = \begin{cases} 0.5 & t+f=2 \text{ のとき} \\ \frac{1-f}{2-t-f} & \text{それ以外} \end{cases}$$

で与えられる.

各統合値は, 図4に示すように, 図的な計算によって

- $I_1(t, f)$: 観測点 (t, f) と点 $(0, 0)$ を結ぶ直線と, 図中の対角線の交点の「次元1」の値
- $I_2(t, f)$: 観測点 (t, f) から図中の対角線に垂線を下ろした点における「次元1」の値
- $I_3(t, f)$: 観測点 (t, f) と点 $(1, 1)$ を結ぶ直線と, 図中の対角線の交点の「次元1」の値

として求められる. 本稿では, 逆転項目平均法に基づく統合値 $I_2(t, f)$ が用いられる.

2.1.3 矛盾度

統合値と並ぶ重要な値が, 矛盾度である. FCR 法では, 人格心理学の分野でしばしば扱われる「好きだが嫌い」といった矛盾した応答や「好きでも嫌いでもない」のような無関心も許容している. そうした応答の内部矛盾は, 観測点の矛盾度(矛盾度指数 あるいは 矛盾-無関連指数とも呼ぶ)として表される. 本稿では, たとえば「あるメッセー

ジには、信頼できる点があるけれども、同時に疑わしい点も見られる」といった状況を扱うために、矛盾度を用いる。

矛盾度 C は、完全な矛盾の場合は $C = 1$ ，数値真理値モデルのような $t + f = 1$ が成り立つときは $C = 0$ ，好きでも嫌いでもないという無関心な状態については $C = -1$ となるように定義される。ひとつの定義として、観測点 (t, f) に関する定和からのずれ

$$C(t, f) = t + f - 1$$

が用いられる。

2.2 Marsh と Dibben によるトラストの分類

トラストの評価値（以下、トラスト値）は、区間 $[-1, 1]$ 上の 1 点である。Marsh と Dibben は具体的なトラスト値の計算方法も与えているが、これについては、文献 [8] などを参照されたい。本稿では、トラスト値の計算方法には依存せず、求められたトラスト値を直接扱うものとする。

Marsh と Dibben は、トラスト値に応じて、次の 4 つの分類を示している。

- **トラスト（信頼, trust）**: 相手あるいは事柄について、信頼を置いている状態。相手あるいは事柄に関するトラストの評価値が 1 に近い状態として定式化する。具体的には、トラスト値が「信頼の閾値」と呼ぶ基準値を越える場合にトラストしていると考えられる。
- **ディストラスト（不信, distrust）**: 相手や事柄について、信頼できないと思っている状態。評価値が負の状態として定式化する。
- **アントラスト（無信頼, untrust）**: 相手や事柄について、信頼して良いかわからない状態。評価値が非負だが信頼の閾値よりも低い状態として定式化する。
- **ミストラスト（誤信頼, mistrust）**: 当初の信頼・予想を（いい方向も含め）裏切られた状態。

これらの分類のうち、ミストラストは信頼の時系列的な変化に関する性質と言える（4 節にて議論する）。トラスト・ディストラスト・アントラストについては、3 節で扱う。

3. 信頼と不信に基づくトラストの二次元表現

Marsh と Dibben のトラスト値に基づき、たとえば

- トラスト値が 0.9
- 信頼の閾値が 0.85

であるような場合を考えよう。このときは、トラスト値が信頼の閾値を越えており、信頼している状態である。しかしながら、不信感が全くないとは言えるだろうか。トラスト値の満点は 1 であるから、トラスト値が 0.9 のときは、その満点までは 0.1 だけ足りないことになる。足りない分だけ、対象を信頼しきれていないとの解釈も可能に見える。

本研究では、こうした解釈ができるのは、一次元的な基準に基づくトラスト値の表現力の不足によるものと考え

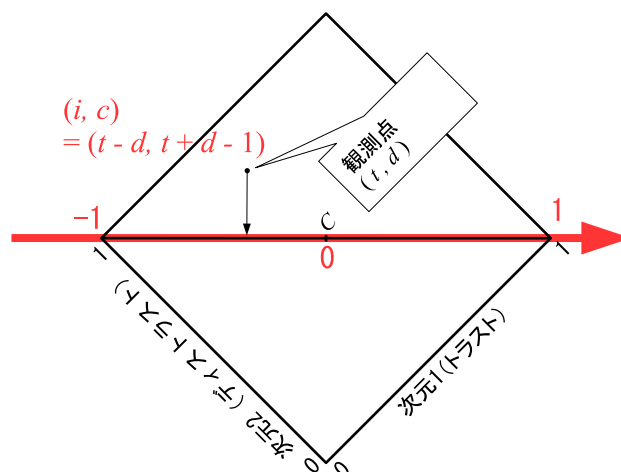


図 5 2 次元的なトラスト値の表現から 1 次元的な表現への変換

た。そこで、信頼の度合いの集合 $Trust$ と不信の度合いの集合 $DisTrust$ を

$$Trust = DisTrust = \{v \mid 0 \leq v \leq 1\}$$

と定め、トラスト値を $Trust \times DisTrust$ の要素（以下では、FCR 法での用語と同じく、観測点と呼ぶことにする）として二次元的に定めるものとする。

3.1 統合値に基づく二次元的なトラスト値の意味

二次元的なトラスト値の意味を考えるため、いくつかの観測点に注目してみよう。(1, 0) 付近にある観測点 $(t, d) \in Trust \times DisTrust$ は信頼 t が高く不信 d が低い「トラスト」の状態を、(0, 1) 付近の観測点は信頼が低く不信が高い「ディストラスト」の状態を表すとみなせる。また (1, 0) と (0, 1) を結ぶ直線上の点は、信頼と不信が定和尺度 ($t + d = 1$) を満たす意味で、理想化された状態と言える。我々は、Marsh と Dibben のトラスト値は、この直線上の点に対応すると考えた。すなわち、従来のトラスト値は理想的な条件に限定して定義されたものである。なお、トラスト値 0 に対応する観測点は (0.5, 0.5) である。

定和尺度 ($t + d = 1$) を満たす観測点について、Marsh と Dibben の分類と同様に、トラストの概念を分類してみたい。信頼の閾値に対応するある正の値 $0 < \alpha < 1$ に対して、点 (1, 0) から点 $(\alpha, 1 - \alpha)$ までの直線上の値は、トラストにあたる観測点とみなせる。じつは、この後に示す「定和尺度を満たすとは限らない観測点」に関する議論から、Marsh と Dibben による信頼の閾値 CT に対して、 α の値は $\alpha = \frac{1 + CT}{2}$ で与えられる。つまり、トラストにあたる観測点は、 $(\frac{1 + CT}{2}, \frac{1 - CT}{2})$ と (1, 0) を結ぶ点である。ディストラストにあたる観測点は、(0.5, 0.5) と (0, 1) を結ぶ点である。実際、これらの観測点では $t < d$ を満たし、不信が信頼を上回っている。 $t + d = 1$ 上のそれ以外の観測点は、アントラストである。

次に、 $t + d = 1$ 上に限らない場合について、上記を一般化してみよう。(1, 0) と (0, 1) を結ぶ直線が区間 $[-1, 1]$ に対応するよう、観測点 $(t, d) \in Trust \times DisTrust$ を

$$\begin{aligned} & \left[\begin{pmatrix} \cos \frac{\pi}{4} & -\sin \frac{\pi}{4} \\ \sin \frac{\pi}{4} & \cos \frac{\pi}{4} \end{pmatrix} \left\{ \begin{pmatrix} t \\ d \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\} + \begin{pmatrix} \frac{\sqrt{2}}{2} \\ 0 \end{pmatrix} \right] \times \frac{1}{\frac{\sqrt{2}}{2}} \\ &= \begin{pmatrix} t-d \\ t+d-1 \end{pmatrix} \end{aligned}$$

で変換する(図5も参照されたい)。移った先 $(t-d, t+d-1)$ を (i, c) と呼ぶ。まず、移った先の第1要素 $i = t-d$ を考えよう。これは $-1 \leq i \leq 1$ を満たす値で、Marsh と Dibben のトラスト値に対応する。実際、値 i は「信頼できると考えている度合い t から不信感の度合い d を引いた差分が、正味のトラスト値である」ことを示しており、直感にも合致する。Marsh と Dibben による信頼の閾値 CT に対して $i = t-d \geq CT$ であればトラスト、 i の値が負であればディストラスト、 $0 \leq i < CT$ であればアントラストである。この基準は、上記の定和尺度に限定した場合にも当てはまる。

図5から、値 i は、観測点 (t, d) から (1, 0) と (0, 1) を結ぶ直線に下ろした垂線の足元の点を求め、さらにとりうる値の区間を $[-1, 1]$ に正規化した値だとわかる。この定め方と図4の図的な統合値の計算方法から、値 i は、FCR法の「逆転項目平均法」による統合値(「トラスト」「ディストラスト」それぞれの度合いから求めた、真の「トラスト」の度合い) $I_2(t, d)$ を、区間が $[-1, 1]$ になるよう正規化したものとみなせる。

3.2 矛盾度によるアントラストのさらなる分類

逆転項目平均法に関して、(1, 0) と (0, 1) を結ぶ直線に対する同じ垂線上の観測点は、同一の統合値をとる。たとえば、観測点 $A = (t, d)$ と下ろした垂線の足元の点

$$A' = \left(\frac{t + (1-d)}{2}, 1 - \frac{t + (1-d)}{2} \right)$$

は同一の統合値をとり、さらには、区間 $[-1, 1]$ 上の同じ値 $i = t-d$ に移る。しかし、観測点 A と A' では、(1, 0) と (0, 1) を結ぶ直線(つまり、定和尺度というある種の理想的な条件を満たすトラスト値の集合)との近さの違いがある。観測点 A とこの直線との距離は、図5より、点 (i, c) の第2要素 $c = t+d-1$ の絶対値として与えられる。値 c は、対象に対して「信頼の度合い t と不信の度合い d の両方が高い」という矛盾した評価を抱いている場合は1に近くなり、逆に対象に対して「信頼の度合いも不信の度合いも低い」という、すなわち信頼も不信も抱いていない無関心の場合は-1に近くなる。これは、2.1.3節で示したFCR法の矛盾度の特徴と一致する。実際、値 c は矛盾度と同一の式で与えられる。

ここで、アントラストのとき、すなわち第1要素 i の値が $0 \leq i < CT$ の場合について考えよう。アントラストは、対象に対する信頼/不信に関する判断がはっきりつけられない状態と言える。しかし、ひとくちにアントラストと言っても、「信頼感と不信感を同時に抱いているために、信頼についての判断ができない」と「対象のことをよく知らないで、信頼についての判断ができない」ことは、大きく異なる。そこで矛盾度の値を用いて、アントラストに対するさらなる分類を行いたい。分類としては、たとえば

- アントラスト(矛盾タイプ): 信頼も寄せられているが同時に不信感も持っている、混乱した状態。 $0 \leq i < CT$ かつ $c \geq 0$ のとき。
- アントラスト(無関心タイプ): 信頼も不信も抱いていない状態。 $0 \leq i < CT$ かつ $c < 0$ のとき。

が考えられる。文献[8]は、アントラストを上記の「無関心」に対応するものとして捉えている。これに対し本稿のFCR法に基づくトラストの定義は、矛盾という観点からのアントラストの存在を新たに示している。

4. トラスト値の時系列的な変化

トラストの4分類のうち、「ミストラスト」は、当初の信頼を(いい方向への裏切りも含め)裏切られたことを表す概念である。これは、時間経過に伴うトラスト値の(短期間で大幅な)変化に関する性質と考えられる。こうしたトラスト値の時系列的な変化を分析・予測できるようにすることは、大規模災害時における被災者やボランティア間におけるトラスト・協力関係の構築[3], [4], [10]を行う上でも重要である。関連して、短期間のうちに構築する集団のトラストについては、swift trust[9], [11]と呼ばれる概念が知られている。本節では、前節の観測点を状態と捉え、その時系列的な変化を状態遷移機械でモデル化する。

4.1 状態遷移機械とトラスト値

状態遷移モデルに基づく分散アルゴリズムの理論として、LynchらのI/O-オートマトン[6], [7]が知られている。I/O-オートマトンの理論では、システムを、イベントを発生しながら状態遷移し続ける機械のあつまりとみなす。イベントのうち、あるものは外部から観測可能であり(そのようなイベントの具体例として、キーボード入力、画面出力、インターネット上での通信などが挙げられる)、またあるものは観測不可能である(イベントの具体例として、計算機内部の処理や、外部から観測できない専用線を用いた通信などが挙げられる)。

I/O-オートマトン X は、アクションの集合 $sig(X)$ 、状態の集合 $states(X)$ 、初期状態の集合 $start(X) \subset states(X)$ 、および遷移の集合 $trans(X) \subset states(X) \times sig(X) \times states(X)$ の組とする。本稿では、終了状態については仮定しない。 $s \xrightarrow{a}_X s'$ は $(s, a, s') \in trans(X)$ を、すなわ

ち、状態 s からアクション a によって状態 s' に遷移することを表す。本稿では、オートマトンの状態は、ある $n \geq 1$ に関して n -項組として与えられるとする。また、 n -項組の各要素には相異なる変数名が割り当てられ、そのアクセス関数が用意されているものとする。このようなモデル化は I/O-オートマトン理論では標準的で、I/O-オートマトンに基づく分散システムの仕様記述言語 IOA [2] でもこうしたモデル化が行われている（付録 A.1 節の記述例と $states$ -部に関する説明も参照されたい）。

本稿では、状態を表す n -項組のなかにトラスト値とディストラスト値のための要素が存在し、そのアクセス関数は、それぞれ $.tr$ および $.dis$ であるとする。たとえば、状態 $s \in states(X)$ におけるトラスト値は $s.tr$ 、ディストラスト値は $s.dis$ である。

4.2 トラストに関する安全性と活性

初期状態から始まる外部から観測可能なイベントを並べた列を、トレースと呼ぶ。トレースは、外部から見えるシステムの「振舞い」を表している。一般にシステムは複数個の（場合によっては無限個の）トレースを持ち、オートマトンの性質はトレースの集合によって特徴づけられる。

トレース集合を用いて定義される分散アルゴリズムの主な性質としては、安全性や活性が知られている。安全性は「悪い振舞いが発生しない」ことを表す性質で、たとえば「0 による除算が起きない」「ある状態 s に達した後、エラー状態 e には到達しない」などがこれにあたる。一方、活性は「望まれる振舞いが、いつか最終的に発生する」ことを表す性質で「ある手続きが、必ず計算結果を返す（プログラムが止まる）」「どの状態からも必ずある状態 s に到達できる」などである。なお、安全性の例として示した「0 による除算が起きない」は、活性ではない。

3 節で示した観測点を状態と考えたとき、たとえば「ディストラストの状態に至らない」「いちどディストラスト以外の状態に達した後、再びディストラストの状態に至ることはない」という性質は、トラストに関する安全性とみなせる。また、「いつかトラストの状態に達する」は、活性とみなせる。

4.2.1 安全性

以下では、 $0 < CT \leq 1$ を信頼の閾値とし、トラストの領域 $T(CT)$ とディストラストの領域 D 、およびアントラストの領域 $U(CT)$ を

$$\begin{cases} T(CT) &= \{(t, d) \mid t \in Trust \wedge d \in DisTrust \\ &\quad \wedge t - d \geq CT\} \\ D &= \{(t, d) \mid t \in Trust \wedge d \in DisTrust \\ &\quad \wedge t < d\} \\ U(CT) &= Trust \times DisTrust \setminus (T(CT) \cup D) \end{cases}$$

とする。

状態 s から状態 s' に到達可能であることを、述語

$$\begin{aligned} &reachable(s, s') \\ \iff & \\ &s = s' \\ &\vee \exists s'' \in states(X) \exists a \in sig(X) \\ &\quad [s \xrightarrow{a}_X s'' \wedge reachable(s'', s')] \end{aligned}$$

で表し、述語 $nonDistr(s)$ を

$$\begin{aligned} &nonDistr(s) \\ \iff & \\ &\forall s' \in states(X) [reachable(s, s') \\ &\quad \implies (s'.tr, s'.dis) \notin D] \end{aligned}$$

で定める。このとき、安全性に関する性質である「ディストラストの状態に至らない」は、 $\forall s \in start(X) [nonDistr(s)]$ と表せる。また、「いちどディストラスト以外の状態に達した後、再びディストラストの状態に至ることはない」は、

$$\begin{aligned} &\forall s \in states(X) \forall s' \in states(X) \\ &\quad [(reachable(s, s') \wedge (s'.tr, s'.dis) \notin D) \\ &\quad \implies nonDistr(s')] \end{aligned}$$

と書ける。

4.2.2 活性

活性に関するトラストの性質を考えることもできる。たとえば、任意の自然数 $n \in \mathcal{N}$ に関して、 n ステップで状態 s から状態 s' に到達可能であることを表す述語 $reachable^n(s, s')$ を

$$\begin{aligned} &reachable^0(s, s') \iff s = s' \\ &reachable^{k+1}(s, s') \\ \iff & \exists s'' \in states(X) \exists a \in sig(X) \\ &\quad [reachable^k(s, s'') \wedge s'' \xrightarrow{a}_X s'] \end{aligned}$$

で与える。このとき「どのような状態からでも（各状態において適切な発火可能なアクションをうまく選べば、）有限ステップでトラストの状態に達することができる」ことは、

$$\begin{aligned} &\forall s \in states(X) \exists s' \in states(X) \exists n \in \mathcal{N} \\ &\quad [reachable^n(s, s') \wedge (s'.tr, s'.dis) \in T(CT)] \end{aligned}$$

と書ける。

4.3 トレース包含に基づくトラストの証明法

トラストに関する安全性や活性の証明については、4.2.1 節や 4.2.2 節で示した論理式を直接的に証明する方法が考えられる。しかしこのほかにも、I/O-オートマトン理論の結果を使った、次のような証明も可能である。

図 6 は、I/O-オートマトンに基づく仕様記述言語 IOA（文献 [2] . A.1 節にも概略を示す）を用いて、ある行為者に対するトラスト値（すなわち、観測点）の移り変わり

```

automaton tester
signature
  output move(ev:Event, pt: VL, pd: VL,
              dt: VL, dd: VL)

  output inDistr
  output nonDistr

states
  tr: VL := 0,    % VL ranges over [-1, 1]
  dis: VL := 0,
  stateOfAgent: State := initState

transitions
  output move(ev, pt, pd, dt, dd)
  pre   pt = tr
        /\ pd = dis
        /\ (0 <= pt + dt /\ pt + dt <= 1)
        /\ (0 <= pd + dd /\ pd + dd <= 1)
        /\ condition(stateOfAgent,
                     ev, pt, pd, dt, dd)
  eff   tr := tr + dt;
        dis := dis + dd;
        stateOfAgent := change(stateOfAgent, ev)

  output inDistr
  pre   tr < dis
  eff   do nothing

  output nonDistr
  pre   ~(tr < dis)
  eff   do nothing

```

図 6 IOA 仕様記述言語による記述例 (オートマトン tester)

を状態遷移機械 *tester* として記述した例である。ここで、*tr* と *dis* はそれぞれ信頼と不信の度合い (値の範囲は $[0, 1]$) を、*stateOfAgent* は行為者の状態を表す (なお、行為者の初期状態は、項 *InitState* を用いて表す) 。またアクション *move*(*ev*, *pt*, *pd*, *dt*, *dd*) は、「ある時点でこの行為者のトラスト値が観測点 (*pt*, *pd*) にあるときにイベント *ev* が起きたとして、その後、トラスト値が観測点 (*pt* + *dt*, *pd* + *dd*) に移ること」を表す。アクション *inDistr* は $(tr, dis) \in D$ のとき、アクション *nonDistr* は $(tr, dis) \notin D$ のときに、それぞれ発火可能である。また、これらのアクションが発火した際には、現状態に遷移する (つまり、任意の状態 $s \in states(tester)$ に関して、 $s \xrightarrow{inDistr} tester\ s$ または $s \xrightarrow{nonDistr} tester\ s$ のどちらかである) 。これらのアクションは、各状態が不信の状態か否かをトレース上に反映させるために使われる。

オートマトン *tester* の振舞いは、アクション *move* の *pre*-部に現れる述語 *condition* の定義による。たとえば、

$$condition(stateOfAgent, ev, pt, pd, dt, dd) \iff (pt, pd) \notin D \wedge (pt + dt, pd + dd) \notin D$$

と定めるときは、(初期状態における (tr, dis) の値が $(0, 0)$ であることに注意すると) オートマトン *tester* のあらゆるトレースは、アクション *inDistr* の出現を含まない。つまり、オートマトン *tester* のトレース集合 $traces(tester)$ が「ディストラストの状態に至らない」という条件を表す

ことになる。実際、オートマトン *tester* が 4.2.1 節の条件 $\forall s \in start(tester) [nonDistr(s)]$ を満たすことは、容易に示せる。

任意のオートマトン *A* およびトレース集合 $traces(A)$ に関して、トレース包含

$$traces(A) \subseteq traces(tester)$$

が成り立つとき、オートマトン *tester* のトレース集合で定められる安全性が、オートマトン *A* でも成り立つ。よって、 $\forall s \in start(A) [nonDistr(s)]$ を直接証明しなくても、オートマトン *tester* とのトレース包含を示せば十分である。

I/O-オートマトンの理論では、フォワードシミュレーションと呼ばれる二項関係を見つける方法をはじめとする、トレース包含を効率的に証明するさまざまな手法が知られている [7]。I/O-オートマトン *Conc* から I/O-オートマトン *Abst* へのフォワードシミュレーション $f \subset states(Conc) \times states(Abst)$ とは、

- (1) 任意の初期状態 $a \in start(Conc)$ に関して、 $f(a, b)$ を満たす初期状態 $b \in start(Abst)$ が存在する。
- (2) 任意の状態 $a_1, a_2 \in states(Conc)$ 、 $b_1 \in states(Abst)$ および動作 $\pi_{Conc} \in sig(Conc)$ に関して、 $f(a_1, b_1)$ かつ $a_1 \xrightarrow{\pi_{Conc}}_{Conc} a_2$ のとき、 $f(a_2, b_2)$ を満たすある状態 $b_2 \in states(Abst)$ およびトレース tr_{Abst} が存在して、 $b_1 \xrightarrow{tr_{Abst}}_{Abst} b_2$ かつ $trace(\pi_{Conc}) = tr_{Abst}$ である。

を満たす二項関係である (ただし $s \xrightarrow{\alpha}_X s'$ は、状態 *s* から始まり状態 *s'* で終わる *X* の実行フラグメント (部分的な実行列) で、 α はその実行フラグメントに現れる外部から観測可能なイベントを順に並べた列である) 。I/O-オートマトン *Conc* から *Abst* へのフォワードシミュレーションが存在するとき、 $traces(Conc) \subseteq traces(Abst)$ が成り立つ。以上のことから、「ディストラストの状態に至らない」という安全性に関するトラストの条件を示すには、オートマトン *A* からオートマトン *tester* へのフォワードシミュレーションをひとつ見つけることにより、トレース包含 $traces(A) \subseteq traces(tester)$ を示せば、十分である。

5. むすび

本稿では、信頼と不信の二軸に基づいてトラスト値を扱った。我々のトラスト値の妥当性は、Marsh と Dibben による一次元的なトラスト値への対応と、FCR 法の矛盾度の考え方に基づく。とくに、矛盾度の考え方を用いることにより、アントラストのさらなる分類 (矛盾タイプと無関心タイプ) が可能となった。

さらに本稿では、時系列的なトラスト値の変化に関する分析を行うため、状態遷移機械に基づくモデル化を行い、観測点の変化に関する性質の基本的な検討を行った。我々は、分散アルゴリズム理論における安全性や活性の観点か

らトラスト値の変化に関する性質を扱った。とくに、安全性に関しては、トレース包含に基づく証明法を検討した。

今後の重要な課題として、災害時に交換されるメッセージのトラスト分析など、より具体的な事例に対する分析を行い、本稿の手法の適用可能性を示すことが挙げられる。

謝辞 本研究は、NICT 委託研究「スマートコミュニティを支える高信頼ネットワーク構成技術の研究開発（副題：ソーシャルメディア時代の高信頼災害時通信の研究開発，管理番号 19303）」により行われた。

参考文献

- [1] 大阪地震、SNS 利用急増、発生直後、LINE メッセージ 5 倍。日経産業新聞，p. 3，2018 年 6 月 19 日。
- [2] Andrej Bogdanov. Formal verification of simulations between I/O-automata. Master's thesis, Massachusetts Institute of Technology, 2000.
- [3] Maria Grazia Busa, Maria Teresa Musacchio, Shane Finnan, and Cilian Fennell. Trust-building through social media communications in disaster management. In *Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion*, pp. 1179–1184, New York, NY, USA, 2015. ACM.
- [4] F. Lemieux. The impact of a natural disaster on altruistic behaviour and crime. *Disasters*, Vol. 38, No. 3, pp. 483–499, July 2014.
- [5] R. J. Lewicki, D. J. B. McAllister, and R. J. Bies. Trust and distrust: New relationships and realities. *Academy of Management Review*, Vol. 23, pp. 438–458, 1998.
- [6] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, 1996.
- [7] Nancy Lynch and Frits Vaandrager. Forward and backward simulations — part I: Untimed systems. *Information and Computation*, Vol. 121, No. 2, pp. 214–233, September 1995.
- [8] Stephen Marsh and Mark R. Dibben. Trust, untrust, distrust and mistrust – an exploration of the dark(er) side. In *Proceedings of the Third International Conference on Trust Management, iTrust'05*, pp. 17–33, Berlin, Heidelberg, 2005. Springer-Verlag.
- [9] Debra Meyerson, Karl E. Weick, and Roderick M. Kramer. *Swift Trust and Temporary Groups in Trust in Organizations: Frontiers of Theory and Research*. SAGE, 1995.
- [10] Yuko Murayama. Issues in disaster communications. *Journal of Information Processing*, Vol. 22, No. 4, pp. 558–565, 2014.
- [11] Jessica Wildman, Marissa Shuffler, Elizabeth Lazzara, Stephen Fiore, and Shawn Burke. Trust development in swift starting action teams: A multilevel framework. *Group & organization management*, Vol. 37, No. 2, pp. 137–170, 2012.
- [12] 小田 哲久. ファジィ多項目並列評定法の特徴について。日本経営システム学会誌，Vol. 12, No. 1, pp. 23–32, 1995.
- [13] 小田 哲久. 多次元多値論理体系の提案：ファジィ論理の多次元的拡張としての超論理空間モデル。日本経営工学会論文誌，Vol. 49, No. 3, pp. 135–145, 1998.
- [14] 小田 哲久. 評定尺度法へのファジィ論的アプローチ — FCR 法，IR 法について —。心理学評論，Vol. 56, No. 1, pp. 67–83, 2013.
- [15] 小田 哲久. 人間工学のための計測手法 第 3 部：心理計測と解析 (3)。人間工学，Vol. 51, No. 5, pp. 293–303, 2015.

- [16] 日本ファジィ学会. ファジィとソフトコンピューティングハンドブック. 共立出版, September 2000.

付 録

A.1 IOA 仕様記述言語

IOA 言語は、状態機械 (I/O-オートマトン) に基づく仕様記述言語である。IOA では、プロセス記述はアクションの前提 (pre-部) とイフェクト (eff-部) から成る。アクションには、

- 内部遷移: 通常の状態機械の遷移 (internal 型を持つ)
 - 出力・入力: 外部から観測可能なアクション (外部アクションと呼ぶ。output 型 もしくは input 型を持つ)
- がある。以下に、IOA 言語によるプロセス記述例を示す。

例 1 プロセス i から j への通信路を表す I/O-オートマトン $\text{ChannelIOA}(i, j)$ 。

$\text{automaton ChannelIOA}(i, j: \text{Int})$

```
signature
  input  send(m: MES, const i, const j)
  output recv(m: MES, const i, const j)
states
  queue: Seq[MES] := empty
transitions
  input send(m, i, j)
    eff queue := m -| queue
  output recv(m, i, j)
    pre len(queue) ~ = 0 /\ last(queue) = m
    eff queue := init(queue)
```

列 Seq やオペレータ (代入 :=, 列の先頭に要素を追加した結果を返す関数 $-|$, 列の長さを返す関数 len , 列の末尾要素を返す関数 last , および、列から末尾要素を取り除いた結果を返す関数 init) は、等式仕様で与えられる。

仕様 $\text{ChannelIOA}(i, j)$ は、次の 3 つの部分から成る。

- (1) signature-部: 内部遷移・入力・出力アクションを宣言するための部分で、上記では入力アクション send と出力アクション recv が宣言されている。
- (2) states-部: 内部変数の宣言である。メッセージの列を問わず内部変数 queue (初期状態は空、型 $\text{Seq}[\text{MES}]$ を持つ) が宣言されている。
- (3) transitions-部: signature-部で宣言した各アクションの定義である。
 - 入力アクション $\text{send}(m, i, j)$: メッセージ m を列 queue の先頭に加え、 queue に再格納する。入力アクションは前提部を持たず、常に発火できる。
 - 出力アクション $\text{recv}(m, i, j)$: 列 queue の長さが 0 でなく、かつ queue の末尾要素が m のとき、発火可能である。列 queue から末尾要素 m を取り除いたものが内部変数 queue に再格納される。