

正しい意見共有に向けたユーザの投稿頻度を考慮した エージェントネットワークシステム： 人とエージェントの関係から人とエージェント集団の関係への 展開

Agent Network System Based on User's Post Frequency for Correct Opinion Sharing: Towards relationship between a person and an agent group from relationship between a person and an agent

山根大輝^{1*} 前川 佳幹¹ 荒井 亮太郎¹ 福本 有季子¹ 佐藤 寛之¹ 高玉圭樹¹
Daiki Yamane¹ Yoshimiki Maekawa¹ Ryotaro Arai¹ Yukiko Fukumoto¹ Hiroyuki Sato¹
Keiki Takadama¹

¹ 電気通信大学

¹ The University of Electro-Communications

Abstract: 本稿では、SNS 上の多数の情報の中から正しい情報を判断することの困難さに着目し、従来手法である Opinion Sharing Model を実世界の SNS に適用可能な形に拡張した意見共有モデル Agent Suggestion System based on Opinion Sharing Model (ASSO) を提案するとともに、正しい意見の共有を促進するエージェントを設計する。具体的には、意見を発信するユーザからなるネットワークと意見を受け取り伝播するエージェントネットワークに分割することで、ユーザの特性(本研究では投稿頻度)を個別に考慮した意見共有が可能となる。実験では、OSM と比較した結果、二つの知見が得られた。まず、(1) ユーザとエージェントを区別しない一層のネットワークシステムである OSM と、ユーザネットワークとエージェントネットワークからなる二層の ASSO との大きな差異はないことが分かった。次に、(2) ユーザから受け取った意見の処理方法の異なる 4 つのエージェントの中で、意見を反射的にエージェントネットワークに共有する設計方法は意見の投稿頻度がユーザ毎に異なる場合でも正しい意見を形成できること、正しい意見をより多く形成できるだけでなく誤った意見を形成する割合は他の設計方法(情報を複数受信してからエージェントネットワークに共有する設計方法)と比べて優位に低いことが示された。

1 はじめに

SNS の発達と普及によって誰でも簡単に情報が発信できるようになった一方で、その手軽さ故に信頼性が精査されないままに情報発信されることが多い。そのような状況では、ネットワーク上に正しい情報と誤った情報が混在し、それを見たユーザの判断に悪影響を与える場合があるため、ユーザが正しい意見を判断出来るシステムが求められている。この問題に対して、ユーザが判断を迷っているときにエージェントが正しい意見を判断し、ユーザをサポートするというアプローチは有効である。例えば、人間の意思決定を抽象化した

Opinion Sharing Model (OSM)[1-3] 上のエージェントは、Autonomous Adaptive Tuning(AAT)[4]に基づいて意見を形成することによって、ソースの情報が 55% 程度の正確性でも 70% 程度の割合で正しい意見を発信できることが示されている。特に、OSM 上においてはセンサエージェントと呼ばれるエージェントが環境からの情報を得るモデル化になっているため、(1) 人とエージェントの切り分けがなされておらず、(2) 環境から受け取る情報の数がセンサエージェント間で同一であり、ユーザの投稿頻度の差異などユーザの特徴を考慮できないという問題がある。

そこで我々は、ユーザネットワークとエージェントネットワークの二層化により、従来モデルで不明瞭であったユーザとエージェントを切り分けた Agent Suggestion System based on Opinion Sharing Model (ASSO)

*連絡先：電気通信大学
住所：〒182-8585 東京都調布市調布ヶ丘 1 丁目 5-1
E-mail: y1710659@edu.cc.uec.ac.jp

を提案するとともに、意見の投稿頻度の異なるユーザの導入を可能にする。また、ASSOではOSM上で正しい意見共有を促進する従来のAATのエージェントが同じ効果を生むか不明であるのでため、そのため、本稿では4つのエージェントの設計方法を提案し、シミュレーション実験を通してASSOの有効性とエージェントの有効な設計方法を検証する。特に、ユーザの意見をネットワークに共有するエージェント(以下、通常のエージェントと区別するためにソースエージェントと呼ぶ)の設計に着目し、それ以外のエージェントはAATエージェントとする(OSMにおけるセンサエージェントとASSOにおけるソースエージェントはなるべく正しい意見を伝搬しようとするという観点で同じ役割を担う)。本稿では2章で従来モデルであるOpinion Sharing Model(OSM)と多数決に基づく重み調整アルゴリズムであるAutonomous Adaptive Tuning(AAT)について説明し、3章でOSMをSNSに適用可能な形に拡張したAgent Suggestion System based on Opinion Sharing Model(ASSO)及びエージェントの設計方法について説明する。また、4章ではASSOがOSMを適切に拡張したことを検証する実験とエージェントの設計方法によるASSO上での正しい意見共有の精度の比較を行った。そして、5章ではASSOにおけるエージェントの設計方法によるエージェントネットワーク全体の効果について考察し、6章で本研究のまとめを行う。

2 Opinion Sharing Model with Autonomous Adaptive Tuning (OSM with AAT)

2.1 Opinion Sharing Model (OSM)

2.1.1 全体像

Opinion Sharing Model(OSM)は、複数のエージェントからなるエージェント集合 A と環境 $\{e\}$ によって構成されるエージェントネットワークモデルである。各エージェントは、自身とリンクで繋がっているエージェント(これを隣人と呼ぶ)とバイナリ値で表される意見 $O = \{\text{white}, \text{black}\}$ を通信する。また、いくつかのエージェントはセンサエージェントと呼ばれ、環境を観測することが出来る(このエージェントは、環境 $\{e\}$ とリンクを持っている)。ここで、観測される情報の正しさは正解情報観測率 r に依存し(反対に確率 $1-r$ で不正解情報が観測される)、ネットワーク上には正しい情報と誤った情報が混在する。エージェントはこれらの情報を受け取ることによって正しいと確信を持った情報を自分の意見として形成し適切に意見を隣人に

伝える。最終的には、ネットワーク全体として多くのエージェントが正しい意見を形成できることを目指す。OSMのイメージを図1に示す。◎はセンサエージェントを示し、○はエージェントを示す。

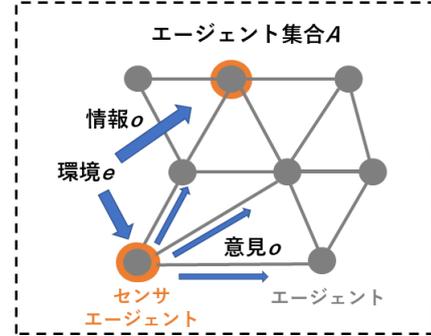


図1: OSMのイメージ

図1では、環境 $\{e\}$ から受信した意見によって意見を形成したセンサエージェントが隣人に意見を伝えている様子を表している。

2.1.2 エージェントネットワーク

ネットワークを構成するエージェント集合 A は以下のように定義される。

$$A = \{i^1 \dots i^{N_A}\}, N_A = |A| \quad (1)$$

ここで、 N_A は総エージェント数、 i は個別のエージェントを表す。次にエージェント i の隣人の集合 D_i を次のように定義する。

$$D_i = \{j^1 \dots j^{d_i}\}, d_i = |D_i| \quad (2)$$

なお、 d_i はエージェント i の隣人数を表し、 j は隣人を表す。また、センサエージェントの集合 S はエージェント集合 A の部分集合であり、 $S \subset A, |S| \ll N_A$ となる。

2.1.3 エージェントの設計

エージェント i は信念値 P_i^k 、意見値 o_i^k 、リンクの重み w_i という3つのパラメータを持つ(k はステップ数と呼ばれるシミュレーション内の時間を表す)。 P_i^k はステップ k において意見をどの程度信じているかを表す指標($0 \leq p_i \leq 1$)であり、隣人 j から受け取った意見 o_j と w_i を用いて更新される(詳細は2.2.2章を参照)。また、 o_i^k はエージェント i がステップ k において形成している意見を表す(詳細は2.2.3章を参照)。エージェント i は P_i^k が閾値 σ (例えば0.9)以上若しくは $1-\sigma$ 以

下に達すると意見を形成する. エージェント i が意見形成の様子を図 2 に示す. 図 2 では, エージェント i がステップ $k-1$ において $P_i^{k-1} = 0.5$ であるとき, 白意見 (white) を二度受け取ることで P_{i+1}^k が σ を超え, 白意見を形成した様子を表している.

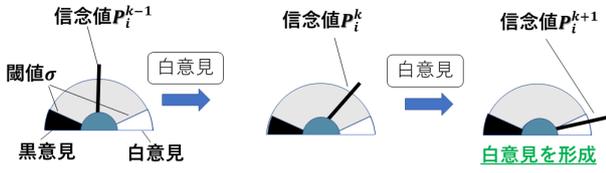


図 2: エージェントの意見形成の様子

なお, エージェントはネットワーク全体の意見形成の様子を把握してはおらず, 自分とリンクの繋がっているエージェントの意見のみを受け取って意見形成する.

2.2 Autonomous Adaptive Tuning (AAT)

2.2.1 エージェントとリンクの重み

リンクの重み w_i はエージェント i が隣人の意見にどの程度影響を受けるかの度合いを表している. つまり w_i の大きさが大きければ早く意見形成し, 小さければ意見形成が遅くなる. この w_i の値が適切でないと, 他人からの少ない意見で自分の意見を決めたり, 他人からの意見を多く待ちすぎて自分の意見を形成できない状況になるため, Autonomous Adaptive Tuning (AAT) は w_i を多数決に基づいた意見共有を可能にする大きさに調整する (エージェントは自身の意見を形成する際, 多数派の意見を参考にしたエージェント集団の多数派の意見を参考にするため OSM with AAT は情報カスケードモデル [6] の一つとなる). この重みは, なるべく多くの隣人から意見を聞いた上で意見を決定できるように調整がとられる. 図 3 に中心のエージェントに 3 人の隣人がいる例を示す. 図中の○はエージェントを表し, 黒塗りの場合『黒』を意見として形成し, 白塗りの場合『白』を意見として形成していることを示す. また■はリンクの重みを表しており, これが大きいほど重みが大きいことを示す.

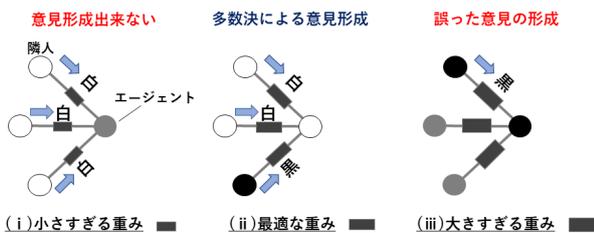


図 3: 重みの大きさによる意見形成の様子の比較

図 3(i) のように小さすぎる重みをリンクに設定すると, 3 人の隣人全ての意見を受け取っても意見形成を行うことが出来ない. 逆に, 図 3(iii) のように大きすぎる重みをリンクに設定すると, 1 人の意見を受け取った時点で意見形成し, たった 1 人の意見を信用する結果になる. この場合, 意見形成は可能であるが情報の正確性の吟味ができず, 誤った意見が拡散されやすくなる. そのため, 図 3(ii) のように隣人に対する共通の重みは小さすぎず, 大きすぎないような最適な大きさの重みに設定する必要がある. AAT は, ハイパーパラメータとして設定した目標意見形成率 h_{tar} に近い意見形成率 (エージェントが意見を形成できた割合) を実現する重みを最適な重みとし, エージェント自身の意見形成率と隣人の意見発信の度合いなどを考慮することでこれを推定する.

AAT の重み推定機構について簡単に説明する. OSM ではエージェント i が受け取る意見の最大数はエージェント i の隣人数 $|D_i|$ に等しいため, リンクに設定する重みの探索範囲を限定することが出来る. そこで, シミュレーションの開始時, 各エージェントは隣人へのリンクに設定する重みの候補集合を生成する. これを W_i と定義し, 式 (3) で表す.

$$W_i = w_i^l : f^l(p_i^0, White, w_i^l) = \sigma, l \in 1 \dots |D_i| \cup w_i^l : f^l(p_i^0, Black, w_i^l) = 1 - \sigma, l \in 1 \dots |D_i| \quad (3)$$

ここで, $f^l(p_i^0, white, w_i^l)$ は, リンクに重み w_i^l を設定した時, エージェント i が white の意見を l 回受け取った時の信念の更新式を表す. W_i 内の重み候補はそれぞれ指定意見数分 ($\leq |D_i|$) だけ意見を受け取ったとき, 意見を形成することが出来る最小の重みとなっている. 全てのエージェントは各ラウンドの終了時, 候補集合の全ての重みに対してその大きさの重みを設定していた場合, 意見形成できたか否かを推定する. そして経過ラウンド数の内, 意見形成できたと推定されるラウンド数の割合を推定意見形成率 \hat{h}_i^l として記録しておく. エージェントが目指す意見形成率である目標意見形成率 h_{tar} を実現する重みに近づく重みを選択する. このことを図 4 を用いて説明する.

重み w	受信意見	受信回数 /	推定意見形成率 $\hat{h}(w_i^l)$
0.91	white	1	0.98
0.87	black	1	0.94
0.70	white	2	0.83
0.64	black	2	0.78
0.55	white	3	0.73
0.49	black	3	0.66

図 4: 重みの候補集合を用いた重み選択

図 4 は $h_{tar} = 0.95$ のときのエージェントが持つ重

みの候補集合の例である。例えば、現在エージェント i が赤枠で囲まれている重み $w = 0.64$ を選択しているとする。この重みは受信意見が black で必要意見数が 2 なので、 $o = \text{black}$ を 2 度受け取ると初めて意見形成できる重みである。ここで、 $h_{tar} = 0.95$ なので、意見形成率 \hat{h}_i^k が 0.94 であると推測される $w = 0.84$ をエージェントは選択したい、しかし、エージェントの意見形成は隣人の意見発信の様子に左右されることから、仮にある一人のエージェントが大きく値の重みに更新した場合、ネットワーク全体の意見伝搬に大きく影響を与えてしまい、各エージェントの重み更新が収束しない可能性がある。そこで、AAT ではネットワーク全体への影響を最小限にするために、時間をかけて少しずつ重みを最適な重みに近づけていくヒルクライミング戦略を用いる。具体的には、各エージェントは初め、重み候補集合の最小の重みを選択する。そして、各ラウンドの終了時、 \hat{h}_i^k を推定し h_{tar} より小さければ一つ大きな重み w_i^{k+1} を選択する。逆に h_{tar} より大きければ一つ小さな重み w_i^{k-1} を選択する。

2.2.2 信念値の更新

エージェント i はステップ k で隣人 j から意見 $o_j \in \{\text{white}, \text{black}\}$ を受信した時、式 (4) の更新式 f に従って信念値 P_i^k を更新する。

$$P_i^k = f(P_i^{k-1}, o_j, w_i) = \frac{wP_i^{k-1}}{(1-w)(1-P_i^{k-1}) + wP_i^{k-1}} \quad (4)$$

where $\begin{cases} w = w_i & \text{if } o_j = \text{white} \\ w = 1 - w_i & \text{if } o_j = \text{black} \end{cases}$

ここで、 P_i^{k-1} は前述したようにステップ $k-1$ でのエージェント i の信念値であり、 w_i はエージェント i から隣人に対するリンクの重みを表す。なお、信念の更新式 f はベイズの定理 [5] に基づいており、 P_i^k は事前確率 P_{i-1}^k に対して $o_j = \text{white}$ を受信した場合の white を信じる事後確率を表す。また、 w_i は尤度を示し、事後確率の大きさは w_i に比例した大きさで更新され P_i^k は 1 に近づく。逆に、エージェント i が $o_j = \text{black}$ を受信した場合 ($o_j = \text{black}$)、 $1 - w_i$ の尤度によって P_i^k は 0 に近づく。シミュレーションでは $k=0$ (初期状態) の時、エージェントはどの意見に対しても確信を持っていないため、初期信念値 P_0^i は 0.5 付近の値をランダムに設定する。

2.2.3 意見の更新と発信

ステップ k において信念値の更新式 f によって信念値が更新されたとき、ステップ k における意見 o_i^k を出

力する意見更新式 F を式 (5)(6) に示す。

$$o_i^k = F(o_i^k, p_i^k, \sigma) \quad (5)$$

$$= \begin{cases} \text{Undeter.} & \text{if } k = 0 \\ \text{white} & \text{if } P_i^k \geq \sigma \\ \text{black} & \text{if } P_i^k \leq 1 - \sigma \\ o_i^{k-1} & \text{otherwise} \end{cases} \quad (6)$$

なお、Undeter はエージェントが意見形成をしていないことを示す。また、エージェントはステップ k において意見形成するとステップ $k+1$ において、意見発信 (意見を隣人に伝える) する。

2.3 OSM with AAT のフロー

OSM with AAT のアルゴリズムを Algorithm 1 に示す。

Algorithm 1 OSM with AAT シミュレーション

```

1: for each Round :  $m \in [0, M-1]$  do
2:   Initialize Beliefs
3:   Initialize Opinions
4:   for each Step :  $k \in [0, K-1]$  do
5:     for each Agent :  $i \in A$  do
6:       (I) if  $i$  is Sensor then
7:         Observe Environment
8:       end if
9:       (II) Receive Neighbor's Opinion
10:      (III) if  $i$  observe or receive any opinions then
11:        Update Belief
12:        Update Opinion
13:        if  $i$  change its opinion then
14:          Send Opinion
15:        end if
16:      end if
17:    end for
18:  end for
19:  Update Weights
20: end for

```

まず、1 行目の Round の繰り返し数を表し、2 行目以下のエージェントの意見共有サイクルを M 回繰り返すことを示している (これをラウンド数と呼ぶ)。2, 3 行目では各エージェントの信念値 P 及び意見値 o を初期化する。4 行目の Step が 1 ラウンド中の経過時間を表しており、ステップ $k=0$ から $k=K-1$ の各ステップにおいて全てのエージェントは (I) 環境観測、(II) 隣人からの意見受信、(III) 信念値更新、意見形成、意見発信を行う。まず、6, 7 行目にて (I) センサーエージェントが環境観測し、環境から情報を受信する。次に、9 行目にて (II) 隣人が意見を発信していれば、意

見を受信する。そして、10-14 行目にて (III) 意見を受信したエージェントは信念値を更新し、意見形成の閾値を超えれば意見を形成して意見発信する。なお、発信した意見は1ステップ後の9行目で観測される。またエージェントは意見を受信しなければ、10-14 行目の処理は何もせずそのステップを終える。そして、19 行目にて、各ラウンド終了時にエージェントの意見形成率を推定し、これが目標意見形成率 h_{tar} に近づくような重みに更新する。

3 Agent Suggestion System based on Opinion Sharing Model (ASSO) と設計

OSM において SNS での意見発信をモデル化するために、環境 $\{e\}$ を SNS ネットワーク上のユーザに置き換え、ユーザが投稿する意見を受信するというモデル化した Agent Suggestion System based on Opinion Sharing Model (ASSO) を提案する。

3.1 ASSO の概要

図5に示すように ASSO は SNS ネットワークを表すユーザネットワーク U と、エージェント同士が意見共有するためのエージェントネットワーク A の2つのネットワークによって構成され、式 (7)-式 (9) のように定義される。

$$U = \{u_1 \dots u_{N_U}\}, N_U = |U| \quad (7)$$

$$A = \{i_1 \dots i_{N_A}\}, N_A = |A| \quad (8)$$

$$|U| = |A| \quad (9)$$

また、任意の x に対してエージェント i_x はユーザ u_x とリンクを持つ。ここで、エージェント i にとって、リンクで接続されたユーザ u を u_i と表記する (逆にユーザ u にとってのエージェント i を i_u と表記する)。このように設定することによって、エージェントはユーザによって発信された意見を直接受け取り、エージェントネットワーク内で共有できる。なお、OSM ではセンサエージェントが環境 $\{e\}$ を観測することで意見をネットワークに取り入れていたが、ASSO ではソースエージェントと呼ばれるエージェントのみが意見をネットワークに共有する。ASSO のイメージを図5に示す。人型シルエットは SNS ユーザ、◎はソースエージェント、○はエージェントを表す。図5では、ユーザが発信した意見 o をソースエージェントが隣人に共有し、その意見を受け取ったエージェントがユーザに正しい意見として提案している様子を示している。

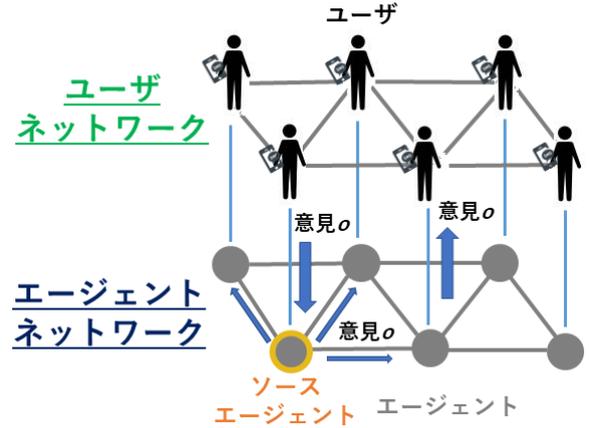


図5: ASSO のイメージ

3.2 ユーザの設計

ユーザネットワークに属するユーザは現実の SNS ユーザに適用可能だが、本稿では第一ステップとして SNS ユーザを意見発信の特性に基づいて以下のパラメータを用いてモデル化する。ここではユーザには以下のパラメータを生成時に設定する。

1. 意見発信率 r_{tweet}
2. 正解意見の発信率 r_{true}

ここで、意見発信率 r_{tweet} はユーザが各ステップにおいて意見発信する確率を表し、正解意見の発信率 r_{true} はユーザが正しい意見を発信する確率を表す。また、各ユーザは全てのステップにおいて Algorithm 2 によって意見発信する。

Algorithm 2 Random Tweet Opinion

```

1: if rand(0,1) <  $r_{tweet}$  then
2:   if rand(0,1) <  $r_{true}$  then
3:     Tweet Truth
4:   else
5:     Tweet Falsity
6:   end if
7: end if

```

具体的には、1 行目で意見発信を行うか否かを判定し、2 行目で正しい意見か誤った意見を発信するかを判定する。この機構によって、ユーザごとの意見の投稿特性を表現可能になる。例えば、意見発信率 $r_{tweet} = 0.01$ で正解意見発信率 $r_{true} = 0.95$ のユーザ A と意見発信率 $r_{tweet} = 0.1$ で正解意見発信率 $r_{true} = 0.2$ のユーザ B の場合、ユーザ A はあまり意見を発信しないが発信する意見はほぼ確実に正しく、逆にユーザ B は意見はよく発信するがよく間違えるユーザとなる。

3.3 エージェントの設計

ソースエージェント以外の通常エージェントの基本的なふるまいは OSM のエージェントと同じであるが、ソースエージェントはユーザに対応するため次の4つのソースエージェントの意見共有方法を提案する。なお、以下の4つの方法を提案する理由は二つある。一つは、ユーザから受け取る意見が少ない場合でもソースエージェントが意見を隣人に伝えられるようにする(方法2-4)ためであり、もう一つはソースエージェントが共有すべき意見をより妥当と考えられる意見を判断してから隣人に伝える方(方法1-3)がネットワーク全体の正しい意見形成につながるのか、ソースエージェントが意見の妥当性を判断しない方が正しい意見形成につながるのかを比較するためである。

方法1: OSM と同じ設計

ユーザからソースエージェントへのリンクの重み w_{iu} を OSM と同じ 0.55 に設定することで、OSM と同様に複数回意見を受け取ることで図2で示すような過程を経て意見形成する。

方法2: ユーザの複数意見に基づいて意見形成し共有する設計

ユーザ u の投稿頻度を考慮して w_{iu} を適切な重みに設定する。具体的にはユーザの期待意見発信数 \hat{c}_i を式(10)を用いて推定し、 $\max\{1, \lfloor \hat{c}_i \rfloor\}$ 回で意見形成できる重みを設定する。

$$\hat{c}_i = \frac{\hat{c}_i m + C_{o_u}^m}{m+1} \quad (10)$$

なお、 m は現在のラウンド数、 $C_{o_u}^m$ はラウンド m においてユーザから受け取った意見の総数を表し、 \hat{c}_i の初期値は 0 である。

方法3: ユーザの意見を全て伝える設計

ソースエージェントは受け取った意見を隣人に発信する。このとき、信念値更新及び意見形成を行わない。

方法4: ユーザの意見更新時に伝える設計

ステップ k で受け取ったユーザの意見がステップ $k-1$ までで受け取った意見の中で最も新しい意見と異なっていればその意見を隣人に発信する。例えば、ユーザが1ラウンド中に black \rightarrow black \rightarrow white と意見発信した場合、ソースエージェントは black \rightarrow white と意見発信する。なお、ユーザから意見を受け取っても信念値更新及び意見形成は行わない。各方法のイメージを図6に示す。この図で用いているアイコンは図5と同じであり、それぞれユーザが白意見や黒意見を発し

た時のエージェントネットワークでの意見伝搬の例を示している。

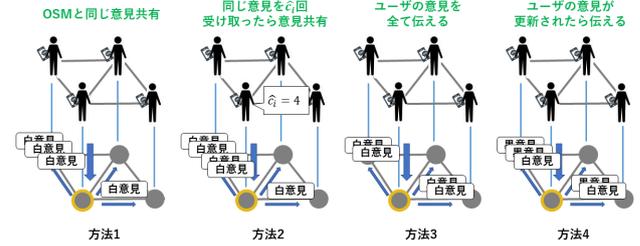


図6: 設計方法によるソースエージェントのふるまいの違い

3.4 シミュレーションフロー

ASSO のアルゴリズムを Algorithm 3 に示す。基本的なフローは OSM with AAT と同じであるため異なる点のみ説明する(差分は赤字で示している)。まず、全てのユーザは各ステップで意見を発信するか否かの判定を行い、発進した場合エージェントはその意見を受け取る(5-8行目)。次に、ソースエージェントは3.3章で述べた方法1-4に従って意見を発信する(10-12行目)。そして、各ラウンドの終了時にユーザの期待意見発信数 \hat{c}_i を計算する(24行目)。

Algorithm 3 ASSO シミュレーション

```

1: for each Round :  $m \in [0, M-1]$  do
2:   Initialize Beliefs
3:   Initialize Opinions
4:   for each Step :  $k \in [0, K-1]$  do
5:     for each User :  $u \in U$  do
6:       opinion  $\leftarrow$  Random Tweet Opinion
7:        $i_u$  receive opinion
8:     end for
9:     for each Agent :  $i \in A$  do
10:      if  $i$  is Sensor then
11:        Opinion Share(Method 1 to 4)
12:      end if
13:      Receive Neighbor's Opinion
14:      if  $i$  observe or receive any opinions then
15:        Update Belief
16:        Update Opinion
17:        if  $i$  change its opinion then
18:          Send Opinion
19:        end if
20:      end if
21:    end for
22:  end for
23:  Update Weights
24:  Estimate  $\hat{c}_i$ 
25: end for

```

4 実験

4.1 実験内容

実験1では、エージェントとユーザを区別しない OSM with AAT と区別する ASSO(方法1)を比較し、実験2ではソースエージェントの有効な設計方法を明らかにするために、OSMと4つの方法を比較する。特に実験1では各センサエージェント及びソースエージェントが情報源(環境 $\{e\}$ 及びユーザを指す)から平均的に受け取る情報数を固定化した場合と各センサエージェント及びソースエージェントによって異なる場合において ASSO がどの程度 OSM with AAT の性能を再現可能かを検証する。実験2では各ユーザの r_{tweet} を変化させた時のネットワーク全体の CR , FR , UR (5.2章で説明)との関係を分析する。

4.2 評価指標とパラメータ設定

評価指標として、エージェント集合の内、正しい意見を形成したエージェントの割合である正意見形成率 CR , 誤った意見を形成したエージェントの割合である誤意見形成率 FR , 意見が形成できていないエージェントの割合である非意見形成率 UR を算出し, CR が高く UR と比較して FR が小さいモデルであるほど性能が高いと評価する。

また、実験1, 2はともに OSM 及び ASSO 上で表1のパラメータを用いた。

表 1: 実験設定

パラメータ	変数	値
ネットワークモデル	—	BA グラフ
ユーザ数	N_U	300
エージェント数	N_A	300
センサエージェント数及びソースエージェント数	N_s	$0.2N_A$
正解意見	o_{true}	white
正解意見発信率	r_{true}	0.55(全てのユーザ)
エージェントの初期信念値	p_{init}	$N(\mu = 0.5, s = 0.1)$ に従う
意見形成の閾値	σ	0.9
ラウンド数	M	300
ステップ数	K	2000
目標意見形成率	h_{tar}	0.95

ここでは一部のユーザのみフォロワーが非常に多いという SNS の特徴を捉えたスケールフリーネットワーク [7] の一つである BA グラフを用いた。また、Prymak[4] によればエージェント数は増加するほど CR の精度が上昇することが報告されているが、本稿では多数決が必ずしも成り立たない状態である少ないエージェント数で高い CR の実現を目指すためユーザ数 N_U 及びエージェント数 N_A を 300 に設定した。なお正解意見は white とし、情報源に悪意がなければ正しい情報の方が誤った情報よりも発信されやすいことを仮定し、誤った情報よりも少しだけ発信率の高い大きい 55% に設定した。

4.3 実験1: OSM with AAT と ASSO(方法1)の性能比較

各センサエージェントが1ラウンドで平均 X 回環境観測するように設定した OSM with AAT と各ユーザの意見発信率 r_{tweet} を X/K に設定した ASSO(方法1)をそれぞれ5シード分実行した。ここで、 r_{tweet} を X/K にする理由は、これによってユーザは1ラウンドの間に平均 X 回の意見発信するので、OSM with AAT の環境観測と同じ役割とみなせるためである。なお、OSM with AAT 及び ASSO(方法1)においてセンサ・ソースエージェントの意見形成に必要な3回をどのエージェントもほぼ必ず受け取れる3.5回から OSM with AAT において70%程度の正意見形成率 CR を実現するのに必要な13.5回までの値の範囲の中で情報源からの情報数による性能差を比較するため、 X を 3.5, 6.5, 10.5, 13.5 とし、それぞれのエージェントネットワーク全体の正意見形成率 CR , 誤意見形成率 FR , 非意見形成率 UR の平均を図7に示す。この図における縦軸は意見形成率、横軸は X を表し、青色、灰色、水色のグラフがそれぞれ OSM with AAT の CR , FR , UR を表し、オレンジ色、黄色、緑色のグラフがそれぞれ ASSO(方法1)の CR , FR , UR を表している。

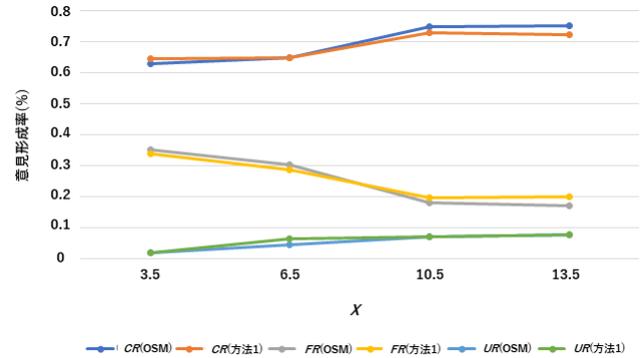


図 7: センサエージェントが環境及びユーザから受信する期待意見数を固定した場合の OSM with AAT と ASSO(方法1)の意見形成率(5シード平均)

図7より X を増加させることによって OSM with AAT と ASSO(方法1)の CR 及び UR は上昇し、 FR は減少する。このことから、センサエージェント及びソースエージェントが情報源から受け取る情報数が増加するほど正解意見を形成しやすくなるだけでなく誤った意見を形成し辛くなるのが分かる。また、OSM with AAT と ASSO(方法1)の CR , FR , UR の差のほとんどない(差の平均は1.36%)。このことからユーザとエージェントを区別しない一層のネットワークシステムである OSM と、ユーザネットワークとエージェントネットワークからなる二層の ASSO との大きな差異

はないことが分かる。次に、ユーザによる投稿頻度の違いを表すために ASSO のユーザの意見発信率 r_{tweet} を $\mathcal{N}(X/K, 1/K)$ に従うように、OSM with AAT のセンサエージェント毎に 1 ラウンドで観測する情報数を $\mathcal{N}(X, 1)$ に従うように設定し、それぞれ 5 シード分実行した。ここで、意見発信数はユーザ毎に異なるものの 1 ラウンド間でソースエージェントが受信する合計意見数の期待値は XN_s 回であり、OSM with AAT においてもセンサエージェント毎に環境観測数は異なるものの合計観測数は XN_s 回となるためほぼ同じ状況を生み出すことができる。 X を 3.5, 6.5, 10.5, 13.5 とした時の結果を図 8 に示す。ここで、図 8 の縦軸・横軸及びグラフの色は図 7 と同様である。

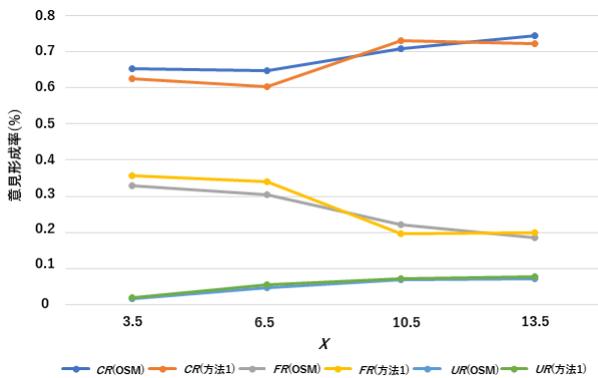


図 8: 情報源から受信する期待意見数がエージェント毎に異なる場合の OSM with AAT と ASSO(方法 1) の意見形成率 (5 シード平均)

図 8 より情報源から受け取る情報数がセンサエージェント及びソースエージェントによって異なる場合でも X を増加させることで図 7 に示す実験と同様の結果が得られることが分かる。また、図 7 と同様 OSM with AAT と ASSO(方法 1) の CR , FR , UR の差のほとんどない (差の平均は 1.95%)。これらより、センサエージェント及びソースエージェントが情報源から受け取る意見数の期待値が同じ場合も異なる場合も、ASSO は OSM with AAT と大きな差異がないこと (2%未満の誤差) から、ASSO において設計方法 1 と他の設計方法を比較することは OSM と比較することとみなす。

4.4 実験 2: 4 種類のソースエージェントの比較

方法 1, 2, 3, 4 に関して、ユーザの意見発信率 r_{tweet} を $\mathcal{N}(X/K, 1/K)$ としてそれぞれ 5 シード分実行した。この結果を図 9-a, 図 9-b, 図 9-c に示す。この図における縦軸はそれぞれ CR , FR , UR を横軸は平均意見発信回数 X を表しており、グラフはそれぞれ青

色が OSM with AAT, オレンジ色が ASSO(方法 1), 灰色が ASSO(方法 2) 色, 黄色が ASSO(方法 3), 水色が ASSO(方法 4) を表している。なお, X は 3.5, 6.5, 10.5, 13.5 である。また, 参考のために実験 1 の 2 つ目の実験で得た, 図 8 に示す OSM with AAT の結果を点線で示している。

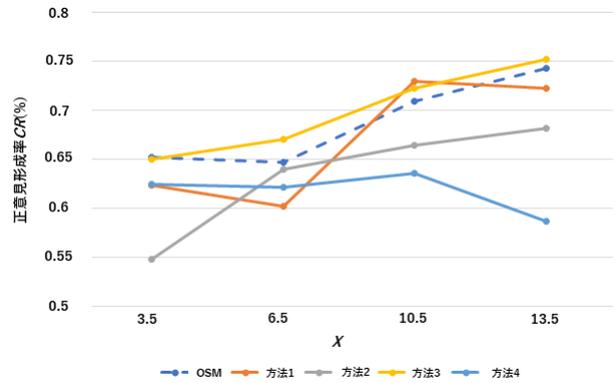


図 9-a: CR の平均の比較

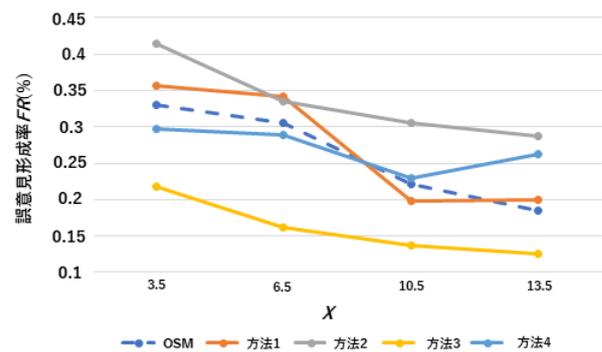


図 9-b: FR の平均の比較

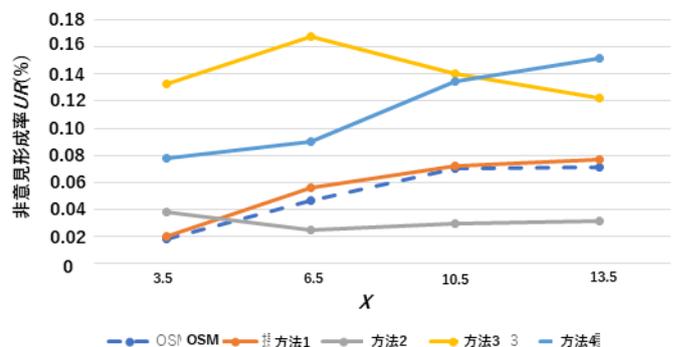


図 9-c: UR の平均の比較

図 9-a, b, c より OSM 及び ASSO(方法 1) は X が大きいほど CR 及び UR が上昇し, これに合わせて FR

が下がっている。環境観測数が多いほどより高い CR が実現される OSM の傾向から、一定数意見形成していないエージェントが存在する方が高い CR を実現することが分かった。また、どの r_{tweet} においても方法3の CR 及び UR が1番もしくは2番目に高い上に FR が常に最も低い。つまり、方法3の設計方法は意見形成しない確率は他と比べて高いが、その分誤った意見を形成することが少なく提案された意見は正しいことが多いといえる。そのため、方法3はユーザに正しい意見を提案することにおいて最も優れている。

5 考察

方法3によるソースエージェントの設計が優位性を持つ理由を考察した後、提案モデルの精度上昇の限界を考察するために r_{tweet} を増加させる分析を行う。

5.1 考察1: 設計方法3の優位性

方法3はソースエージェントがユーザの意見を全て隣人にするものの、各エージェントはすぐに意見を形成せず、たくさんの意見を聞くことで慎重に意見を決めている。このことを確認するために各方法について、1ラウンド内の各ステップにおいて意見を形成していないエージェントの割合(ステップごとの UR) を平均し比較した (r_{tweet} は $\mathcal{N}(6.5/K, 1/K)$ に従うように設定した)。この結果を図10に示す(横軸は設計方法名、縦軸は1ラウンド間の各ステップにおいて UR が占める割合の平均値)。

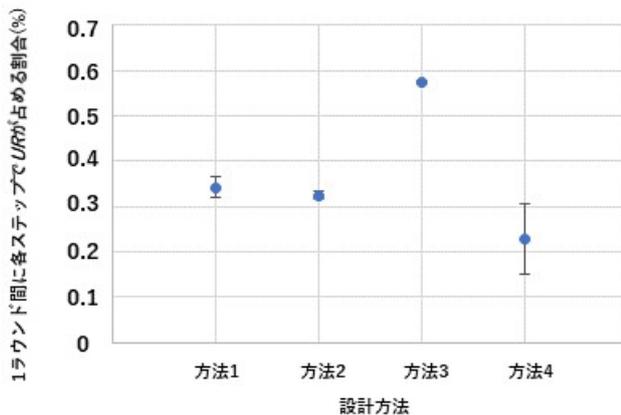


図10: 1ラウンド間に UR の占める割合の平均値と標準偏差 (5シード分)

図10より方法3が最も UR が占める割合が大きいることが分かる (2番目に UR が占める割合の大きな方法2と比較すると1.77倍大きい)。方法3ではソースエー

ジェントはユーザから受け取った意見をそのまま隣人に伝えるため、隣人はユーザが発した正しい意見も誤った意見も多数受け取ることになるが、その分中々意見を形成しにくくなる。しかし、whiteもしくはblackの内、一方の意見を多く受け取った場合にはその意見を形成できる。つまり、隣人の意見が一貫していない場合は意見形成せず、一貫した時は意見形成することで (一貫する意見は正しい意見である可能性は合理的に若干高いため)、正しい意見を形成するエージェントが徐々に増え、やがてネットワーク全体に広がっている。これは、誤った意見を形成することを防ぐことに成功している。また、ソースエージェントがユーザの意見の中から妥当であると判断した意見をネットワークに伝搬させる方法 (方法1-3) は、この時点でユーザに対応するソースエージェントだけで情報の正しさがある程度判断してしまっている。そのため、ネットワーク全体の一部であるソースエージェントの判断が間違っていた場合、ネットワーク全体が誤ってしまう。しかし、方法3ではソースエージェントは何かしらの判断は下さない代わりに純粋に意見をネットワークに意見を伝搬させるため、ネットワーク全体で正しい意見の意見の判断を行うことができる。すなわち方法3はエージェント集団を最も効率よく活用して正しい意見を判断できる方法である。

5.2 考察2: 多数派の逆転現象に対する効果

ユーザネットワークにおいて、各ラウンドの $k=0$ から $k=K-1$ までである意見 o が常に多数派の場合、エージェントネットワークでは意見 o が形成されている確率は高い。しかし、ユーザネットワークにおいて各ユーザが最初に発信した意見の内、多数派だった意見 o と最終的に多数派になった意見 o' が異なる場合はどうであろうか。これを分析するため、 r_{tweet} が $\mathcal{N}(6.5/K, 1/K)$ に従うように設定した ASSO 上で、上記のような場合エージェントネットワークが最終的に意見を正しい意見を形成できた割合を平均し比較した。この結果を図11に示す。横軸は設計方法名、縦軸はエージェントネットワーク全体で正しい意見を形成出来た割合を示す。この図ではユーザが最初に発信した意見の内、多数派だった意見 o が正しく、最終的に多数派になった意見 o' が誤った意見だった場合にエージェントが正しい意見を形成できた割合を青色のグラフが示し (正 → 誤パターン)、最初に発信した意見の内、多数派だった意見 o が誤っていて、最終的に多数派になった意見 o' が正しい意見に逆転した場合にエージェントが正しい意見を形成できた割合をオレンジ色のグラフが示している (誤 → 正パターン)。

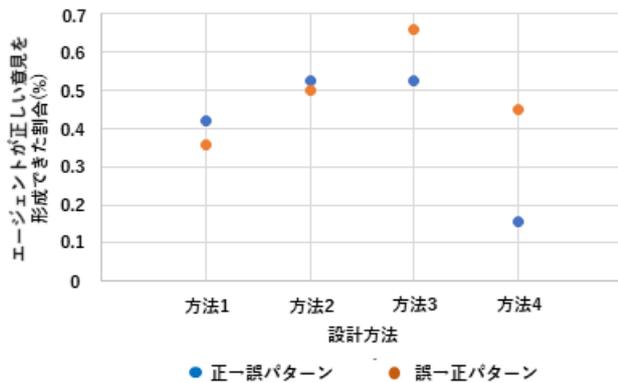


図 11: エージェントが正しい意見を形成できた割合の平均値 (5 シード分)

図 11 より, 設計方法 3 はユーザネットワークが誤った意見から正しい意見に多数派が逆転した場合, それに合わせてエージェントネットワークも正しい意見を形成出来るだけでなく, ユーザネットワークの多数派が正しい意見から誤った意見に逆転した場合は正しい意見に留まる確率が高いことが分かる. これはユーザー層で誤った意見を訂正できる可能性を示唆している.

5.3 考察 3: ASSO の意見発信率 r_{tweet} の増加による精度向上の限界

ASSO における CR の精度上昇の限界を明らかにするために, 意見発信率 r_{tweet} と ASSO 上でのエージェントの正意見形成率 CR の関係を分析する. 実験と同様にユーザの意見発信率 r_{tweet} を $\mathcal{N}(X/K, 1/K)$ とし, 設計方法 1-4 についてそれぞれ 5 シード分実行した. この結果を図 12 に示す.

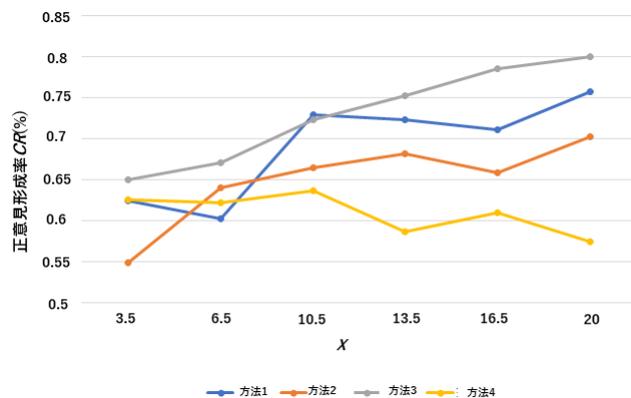


図 12: CR の平均値の比較 (5 シード分)

この図では, 縦軸は正意見形成率 CR を表し, 横軸はユーザの意見発信率 r_{tweet} を表す. また, 青色のグ

ラフは方法 1, オレンジ色のグラフは方法 2, 灰色のグラフは方法 3, 黄色のグラフは方法 4 を示している. なお, X は 3.5, 6.5, 10.5, 13.5, 16.5, 20, 25 としている. 図 12 より, 方法 4 は r_{tweet} が増加するほど CR が低下することが分かり, グラフの傾向から方法 1, 3 は $\mathcal{N}(20/K, 1/K)$ で CR の値はおおよそ収束しており, 特に方法 3 は安定して CR が増加していることが分かる. また, 方法 3 は $X=10.5$ を除き全ての X で最も高い CR を実現しており, その安定さを踏まえると意見数に依らず正しい意見を共有しやすい方法であるといえる.

6 まとめと展望

本稿では, 実際の SNS ネットワークに応用可能な ASSO を提案するとともに, ユーザに正しい意見を提案するためのエージェントを設計し, その有効性を検証した. 実験 1 では, OSM with AAT と ASSO (方法 1) の誤差が 2% 未満であることから, エージェントと人の区別をしていないネットワークと二層化によって区別したネットワークでの性能は同等であることが示され, 実験 2 において設計方法 3 が最も高い CR と低い FR を示すことが分かった. また, 実験及び考察から人から得た情報の正しさをエージェント単体若しくは一部のエージェントである程度判断してからエージェントネットワークに意見共有するのではなく, 得られた意見を純粋に共有しエージェント集団の全員が協力し合うことで高い CR の実現が可能になることが示された. また, 今後の展望としてユーザ毎に r_{true} を異なる値に設定した場合 (発信する意見の正確性が異なる) における正しい意見共有の方法, 発信される意見が少ない話題に対応するために意見発信数が 3.5 よりも少ない場合の意見共有方法及びユーザに悪意がある場合の意見共有方法を議論することなどが挙げられる.

参考文献

- [1] R. Grinton, P. Scerri and K. Sycara : Towards the Understanding of Information Dynamics in Large Scale Networked Systems, *Information Fusion (FUSION)*, pp.794-801(2009)
- [2] R. Grinton, P. Scerri, and K. Sycara : Exploiting scale invariant dynamics for efficient information propagation in large teams, *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pp. 21-30 (2010)
- [3] R. Grinton, P. Scerri, and K. Sycara : An investigation of the vulnerabilities of scale invariant dynamics in large teams, *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pp. 677-684 (2011)

- [4] O. Prymak, A. Rogers, and N. Jennings : Efficient opinion sharing in large decentralised teams, *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems 2012 (AAMAS 2012)*, pp. 543–550 (2012)
- [5] D. Easley and J. Kleinberg : *Networks, crowds, and markets: Reasoning about a highly connected world*, Cambridge University Press (2010)
- [6] S. Bikhchandani, D. Hirshleifer, and I. Welch : A theory of fads, fashion, custom, and cultural change as informational cascades, *Journal of Political Economy*, Vol. 100, No.5, pp. 992–1026 (1992)
- [7] R. Albert and A. Barabási : Statistical mechanics of complex networks, *Reviews of modern physics*, Vol.74, No.1, pp.47-97 (2002)